

Hyperion® Integration Server

Release 2.0



System Administrator's Guide



Hyperion Solutions Corporation

P/N: H70D16-2000000

© 1998–2000 Hyperion Solutions Corporation. All rights reserved.

Hyperion and Essbase are registered trademarks, and Hyperion Solutions is a trademark of Hyperion Solutions Corporation.

Microsoft is a registered trademark, and Windows is a trademark of Microsoft Corporation. IBM, DB2, Lotus, and 1-2-3 are registered trademarks of IBM Corporation. All other brand and product names are trademarks or registered trademarks of their respective holders.

No portion of this manual may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or information storage and retrieval systems, for any purpose other than the purchaser's personal use, without the express written permission of Hyperion Solutions Corporation.

Notice: The information contained in this document is subject to change without notice. Hyperion Solutions Corporation 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, CA 94089

Printed in the U.S.A.

Contents

Preface

Chapter 1: Using OLAP Integration Server

| | |
|--|-----|
| Introducing OLAP Integration Server | 1-1 |
| Starting OLAP Integration Server | 1-2 |
| Shutting Down OLAP Integration Server | 1-6 |
| Viewing the OLAP Integration Server Log File | 1-6 |

Chapter 2: Working with Users, Locks, and Permissions

| | |
|--|-----|
| Working with Users | 2-1 |
| Managing Source Database Users and Hyperion Integration Server Users | 2-2 |
| Working with Locks | 2-2 |
| Using Standard Access Mode | 2-2 |
| Using Exclusive Access Mode | 2-3 |
| Correcting Problems with Locks | 2-4 |
| Viewing Hyperion Integration Server Users with Locks | 2-4 |
| Deleting Locks for Hyperion Integration Server Users | 2-6 |
| Working with Permissions | 2-6 |
| Viewing Permissions for OLAP Models | 2-7 |
| Viewing Permissions for Metaoutlines | 2-9 |

Chapter 3: Troubleshooting ODBC and Connections

| | |
|---|-----|
| Checking Common Connection Problems | 3-2 |
| Database-Specific Problems | 3-4 |
| About ODBC and Connection Problems | 3-4 |

| | |
|--|------|
| Isolating Connection Problems | 3-6 |
| Testing the Database Client Connection to the RDBMS | 3-7 |
| Testing the ODBC Connection to the RDBMS | 3-9 |
| Testing the OLAP Integration Server Connection to the RDBMS | 3-13 |
| Testing the Hyperion Integration Server Desktop Connection to the RDBMS | 3-16 |
| Testing the OLAP Integration Server Connection to Hyperion Essbase Server | 3-17 |
| Testing the Hyperion Integration Server Desktop Connection to Hyperion Essbase Server | 3-19 |
| Correcting Connection Problems | 3-20 |
| User Name and Password Problems | 3-21 |
| Correcting RDBMS Problems | 3-22 |
| Correcting Database Client Problems | 3-24 |
| Correcting ODBC Problems | 3-25 |
| Correcting OLAP Integration Server Problems | 3-27 |
| Correcting Hyperion Integration Server Desktop Problems | 3-28 |
| Using ODBC Tracing | 3-30 |
| Using ODBC Tracing on Windows Systems | 3-30 |
| Using ODBC Tracing on UNIX Systems | 3-31 |

Chapter 4: Tables in OLAP Metadata Catalog

| | |
|--|-----|
| About OLAP Metadata Catalog | 4-2 |
| Tables Relating to the OLAP Model | 4-3 |
| Tables Relating to the Metaoutline | 4-5 |
| Tables Relating to Drill-Through | 4-8 |
| Miscellaneous Tables | 4-9 |

Chapter 5: Using OLAP Command Interface

| | |
|---|-----|
| Starting OLAP Command Interface | 5-2 |
| About OLAP Command Interface | 5-3 |
| Using SET and LOAD Commands | 5-4 |
| Using Informational Commands | 5-5 |
| Using Command Syntax and Execution | 5-6 |
| Recording Member and Data Load Scripts | 5-6 |
| Running OLAP Command Interface Script and Batch Files | 5-7 |
| Running OLAP Command Script Files | 5-8 |

| | |
|---|------|
| Using OLAP Command Interface Commands | 5-9 |
| LOGIN | 5-10 |
| STATUS | 5-10 |
| VERSION | 5-11 |
| SETSOURCE | 5-11 |
| SETTARGET | 5-12 |
| SETCATALOG | 5-12 |
| LOADMEMBER | 5-13 |
| LOADDATA | 5-15 |
| LOADALL | 5-17 |
| LOGOUT | 5-19 |
| SHUTDOWN | 5-19 |
| EXIT | 5-20 |
| Loading Dimensions and Members Using OLAP Command Interface | 5-20 |
| Loading Data Using OLAP Command Interface | 5-21 |

Appendix A: OLAP Integration Server Error Messages

| | |
|---|-----|
| Member Load Error Messages | A-1 |
| Data Load Error Messages | A-5 |
| Drill-Through Report Error Messages | A-6 |
| Miscellaneous Error Messages | A-7 |

Glossary

Index

Preface

Purpose

This guide provides the information you need to configure and maintain Hyperion Integration Server and OLAP Metadata Catalog, and to use OLAP Command Interface on all supported platforms.

Audience

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

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

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

See Microsoft Windows documentation for more information on these terms.

- For UNIX system users, a fundamental understanding of UNIX system administration, such as directory navigation, environment variables, login scripts, and text file editing.

See the documentation for the UNIX system you are using for more information.

- A fundamental understanding of database administration, such as database tables, columns and records, user management, operation of database client utilities, and executing SQL commands.

Document Structure

This document contains the following information:

[Chapter 1, “Using OLAP Integration Server,”](#) describes how to start and stop the server and view the server error log.

[Chapter 2, “Working with Users, Locks, and Permissions,”](#) describes how to view information about users, OLAP models and OLAP metaoutlines, and how to manage access to documents within an OLAP Metadata Catalog.

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

[Chapter 4, “Tables in OLAP Metadata Catalog,”](#) describes the structure and content of the OLAP Metadata Catalog.

[Chapter 5, “Using OLAP Command Interface,”](#) describes the basic operation and commands for the OLAP Command Interface.

[Glossary](#), a list of key Hyperion Integration Server terms and their definitions.

[Index](#), a list of Hyperion Integration Server terms and their page numbers. In the online version, select an index entry to view that page.

Related Documentation

The following documents contain additional related information:

Hyperion Integration Server documents:

- *Hyperion Integration Server Installation Guide*, for information about installation procedures, instructions for connecting relational data sources to OLAP Metadata Catalog, and information about Hyperion Integration Server directories and files.
- *Hyperion Integration Server Desktop OLAP Model User’s Guide*, for more information about basic multidimensional modeling concepts and about how to design, create, and maintain OLAP models based on a relational data source.
- *Hyperion Integration Server Desktop OLAP Metaoutline User’s Guide*, for information about how to design, create, and maintain one or more metaoutlines based on an OLAP model.

Hyperion Essbase OLAP Server documents:

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

Online Help

- To access online help:
 1. In any dialog box, click **Help**.
 2. In the **Help** dialog box, click an item to display information about the item.
- To print an online help topic, display the topic and select File > Print.

Online Guides

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

- ▶ To display an online guide, select Help > Help Topics.

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 text indicates words or characters that you type exactly as they appear on the page. Bold in procedural steps highlights major user interface elements. |
| CAPITAL LETTERS | Capital letters denote commands and various IDs. (Example: CLEARBLOCK command) |
| Example text | Courier font indicates that the material shown is a code or syntax example. |
| Ctrl + 0 | Keystroke combinations shown with the plus symbol (+) indicate that you should press the first key and hold it while you press the next key. Do not type the + symbol. |
| <i>Courier italics</i> | Courier italic text indicates a variable field in command syntax. Substitute a value in place of the variable shown in Courier italics. |
| <i>Italics</i> | Italics in a product-related term indicates that the term is also included in the glossary of the book. |
| Ellipses (...) | Ellipsis points indicate that text has been omitted from an example. |

Table i: Conventions Used in this Document (Continued)

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

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

Additional Support

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

Ordering Documentation

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

Training Services

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

Instructor-led training is delivered in formats and in locations suited to Hyperion's diverse, global customers. Hyperion Authorized Training Centers offer courses that they develop, as well as those developed by Hyperion. Custom training—training on the configured and tailored applications that employees use on the job—is another option to enhance user productivity and to ensure smooth day-to-day operations. Multimedia training—including computer-based training,

Web-based training, and interactive distance learning—provides a cost-effective means of giving users a hands-on introduction to product features and functions. Computer-based training (CBT) provides high-quality interactive training at the user's convenience, regardless of location.

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

Consulting Services

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

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

Technical Support

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

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

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

Using OLAP Integration Server

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

- [“Introducing OLAP Integration Server” on page 1-1](#)
- [“Starting OLAP Integration Server” on page 1-2](#)
- [“Shutting Down OLAP Integration Server” on page 1-6](#)
- [“Viewing the OLAP Integration Server Log File” on page 1-6](#)

Introducing OLAP Integration Server

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

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

OLAP Integration Server must be running if you want to use Hyperion Integration Server Desktop (the graphical user interface) or OLAP Command Interface (the command line interface).

Starting OLAP Integration Server

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

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

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

► To start OLAP Integration Server, use either of the following methods:

- From the Windows Start menu, select OLAP Integration Server.
- Type the executable name on the command line; for example, type:

olapisvr

Note: If OLAP 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 *Hyperion Integration Server Installation Guide*.

When starting OLAP Integration Server from the command line, use the switches specified in [Table 1-1](#) if you want to change the default settings:

Table 1-1: OLAP Integration Server Switches

| Switch | Description |
|--------|---|
| -X | <p>To view a list of available switches for starting OLAP Integration Server, type -x when you start OLAP Integration Server; for example:</p> <pre>olapisvr -x</pre> |
| -C | <p>To set the number of records that OLAP Integration Server commits to Hyperion Essbase during a data load, type -Cnumber of records when you start OLAP Integration Server. This setting specifies the number of records to be committed to Hyperion Essbase after all records are sent. The default setting is all records.</p> <p>The recommended setting for testing purposes is 5,000 to 10,000 records. After you have completed data load testing, shut down the server to reset the number of records to the default setting of all records.</p> <p>Setting a smaller number of records during testing enables you to see data load errors incrementally; for example, after every 5,000 or 10,000 records have been transferred to Hyperion Essbase. Viewing the errors provides you an opportunity to correct data source problems before continuing and waiting for an entire data load to complete before you can take any corrective actions. Resetting the number of records to the default of all records results in faster performance, but error records are viewable only at the completion of the data load.</p> <p>For example, to set the number of records that OLAP Integration Server incrementally transfers to Hyperion Essbase during a data load, type:</p> <pre>-C5000</pre> <p>Note: This setting does not apply to member loads.</p> |

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

| Switch | Description |
|--------|---|
| -E | <p>To give the log file a different name, type -Elog_file_name when you start OLAP Integration Server. The .log extension is added automatically. The default name and location is <i>ISHOME\bin\olapisvr.log</i>.</p> <p>For example, to name the log file <i>myserver.log</i> and put it in the <i>temp</i> directory, type:</p> <p>olapisvr -Ec:\temp\myserver</p> |
| -L | <p>To set the level of detail of the messages that OLAP Integration Server logs, type -Llevel when you start OLAP Integration Server. The default level is 2.</p> <p>For example, to use the highest level of logging, type:</p> <p>olapisvr -L0</p> <p>You can specify the following levels:</p> <ul style="list-style-type: none"> • 0 to log all debug messages and the ODBC SQL generated by OLAP Integration Server • 2 to log all informational messages and some of the ODBC SQL generated by OLAP Integration Server • 3 to log all warnings • 4 to log all non-critical errors • 7 to log all critical errors where OLAP Integration Server aborts the command • 8 to log all critical server errors <p>The recommended level settings are 2 or 3.</p> |
| -M | <p>To set the Hyperion Essbase message database file to a different name, type -Mdatabase_file_name when you start OLAP Integration Server. The default name is <i>ARBORPATH\bin\essbase.mdb</i>.</p> <p>For example, to set the Hyperion Essbase message database file name to <i>essbase6\bin\essbase.mdb</i>, type:</p> <p>olapisvr -Mc:\essbase6\bin\essbase.mdb</p> |

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

| Switch | Description |
|--------|--|
| -T | <p>To set the number of network listeners that OLAP Integration Server starts, type -Tnumber when you start OLAP Integration Server. The default number is 10. The recommended setting is 10.</p> <p>Network listeners receive requests from Hyperion Integration Server client programs. The server automatically adds and subtracts listeners as needed, so the number of listeners set at startup does not limit the number of users that can connect to an OLAP Integration Server.</p> <p>For example, to set the number to 17, type:</p> <p>olapisvr -T17</p> |
| -B | <p>To set the buffer size that OLAP Integration Server uses when transferring rows from the data source to the Hyperion Essbase database, type -BnumberOfRows when you start OLAP Integration Server. The default number is 100 rows.</p> <p>To maximize performance, estimate the size in bytes of the rows being loaded from the relational data source. Divide 64,000 by that estimated size to determine the optimal number of rows to buffer. The goal is to generate blocks of data that are as close to 64 KB as possible, without exceeding that amount. If the size of a block of rows exceeds the 64 KB threshold, the load will fail.</p> <p>For example, to set the number to 150, type:</p> <p>olapisvr -B150</p> |
| -P | <p>To set the TCP port number at which OLAP Integration Server listens, type -PportNumber when you start OLAP Integration Server. This switch is for debugging purposes only.</p> <p>Do not use this switch to set the port number. If you set the port number to a value other than the default, the Hyperion Integration Server client programs will be unable to connect to OLAP Integration Server. The default port number is 3388.</p> |

Shutting Down OLAP Integration Server

- ▶ When you finish using OLAP Integration Server, shut it down by using either of the following methods:
- On Windows NT and Windows 2000 systems, close the OLAP Integration Server window.
 - Issue the SHUTDOWN command in the OLAP Command Interface.
For more information, see [“Using OLAP Command Interface Commands” on page 5-9](#).

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

Note: Always close Hyperion Integration Server Desktop before shutting down OLAP Integration Server.

Viewing the OLAP Integration Server Log File

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

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

- All error messages
- The SQL statements that OLAP Integration Server generates to access the relational data source
- The OLAP Command Interface commands used to perform an action
- Connection information
- Notification of the success or failure of a load
- Other informational messages

For more information on setting the logging level, see [Table 1-1](#).

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

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:

- “Working with Users” on page 2-1
- “Working with Locks” on page 2-2
- “Working with Permissions” on page 2-6

For more information about OLAP Metadata Catalog, see [Chapter 4, “Tables in OLAP Metadata Catalog.”](#)

Working with Users

When working with Hyperion Integration Server, you must manage access for two types of users:

- Hyperion Integration Server users. These users can access OLAP models and metaoutlines stored in OLAP Metadata Catalogs. Hyperion Integration Server users are the database users defined in the relational database that contains the OLAP Metadata Catalog you are using.
- Hyperion Essbase users. These users can access the Hyperion Essbase database that you create from a metaoutline.

Managing Source Database Users and Hyperion Integration Server Users

Source database users are defined in the relational database management system (RDBMS) that manages the source database. Hyperion Integration Server users are defined in the relational database that contains an OLAP Metadata Catalog. Source database users can access the tables and views in the relational source database. Hyperion Integration Server users can access the OLAP models or metaoutlines that you create using Hyperion Integration Server Desktop.

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

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

Working with Locks

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

Using Standard Access Mode

When you open an OLAP model in standard access mode, Hyperion Integration Server gives you a read lock on that OLAP model. When you open a metaoutline in standard access mode, Hyperion Integration Server 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, Hyperion Integration Server upgrades the read lock to a write lock for the duration of the save. After Hyperion Integration Server saves the changes to the OLAP model or

metaoutline, the lock reverts back to a read lock. However, if other users have the OLAP model or metaoutline open, your changes are not updated for the other users until they reopen the document. In addition, if other users have the OLAP model or metaoutline open at the same time as you do, they can edit and save it, thereby overwriting your changes. Using exclusive access mode prevents users from overwriting your changes. For more information, see [“Using Exclusive Access Mode” on page 2-3](#).

- ▶ If you have opened and edited an OLAP model or metaoutline in standard access mode, and are concerned that other users may overwrite your changes, perform one of the following steps:
 - Save the OLAP model or metaoutline using a different name. For more information, see the *Hyperion Integration Server Desktop OLAP Model User's Guide* and the *Hyperion Integration Server Desktop OLAP Metaoutline User's Guide*.
 - If you have not yet started editing the document or have entered only a few changes, close the OLAP model or metaoutline; then, reopen it using exclusive access mode and reapply the changes.
 - Contact the users with read locks and ask them to close the OLAP model or metaoutline. To view a list of users with read locks, see [“Viewing Hyperion Integration Server Users with Locks” on page 2-4](#).

Using Exclusive Access Mode

Opening an OLAP model or metaoutline in exclusive access mode eliminates the risk of other users overwriting your changes. When you open an OLAP model in exclusive access mode, Hyperion Integration Server 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, Hyperion Integration Server 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.

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.

Correcting Problems with Locks

Due to the complex nature of Hyperion Integration Server 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 Hyperion Integration Server Users with Locks” on page 2-4](#). If you are convinced that some or all of the locks are from terminated user sessions, delete them. For more information, see [“Deleting Locks for Hyperion Integration Server Users” on page 2-6](#).

Viewing Hyperion Integration Server Users with Locks

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

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

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

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

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

2. View the results.

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

| CB_USER_NAME | CB_OBJECT_ID | CB_OBJECT_TYPE | CB_OBJECT_LOCK | CB_LOCK_TIMESTAMP |
|--------------|--------------|----------------|----------------|-------------------|
| sa | 889844639 | 1 | 1 | Apr 9 2000 4:43PM |
| sa | 889845263 | 2 | 1 | Apr 9 2000 4:43PM |
| sa | 889844639 | 1 | 1 | Apr 9 2000 5:20PM |
| sa | 892167813 | 2 | 1 | Apr 9 2000 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 Hyperion Integration Server 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 Hyperion Integration Server Users with Locks” on page 2-4](#).

2. Delete the rows containing the unwanted locks.

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

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

Working with Permissions

When you save an OLAP model or metaoutline for the first time, you determine what read or write permissions other users have. Hyperion Integration Server 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 Hyperion Integration Server Desktop. For more information, see the *Hyperion Integration Server Desktop OLAP Model User's Guide*.
 - Save the metaoutline with a different name by selecting File > Save As. For more information, see the *Hyperion Integration Server Desktop OLAP Metaoutline User's Guide*.
 - To edit tables containing the permissions for the OLAP model or metaoutline, use the tools provided with the relational database that contains OLAP Metadata Catalog. For more information on the tables to edit, see [“Viewing Permissions for OLAP Models” on page 2-7](#) or [“Viewing Permissions for Metaoutlines” on page 2-9](#).

Viewing Permissions for OLAP Models

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

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

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

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

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

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

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

2. View the results.

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

| MODEL_NAME | MODEL_OWNER | MODEL_ACCESS_CODE |
|-----------------|-------------|-------------------|
| ----- | ----- | ----- |
| TBC Model | sa | 2 |
| TBC_Mod_Archive | sa | 1 |
| TBC_Mod_Mine | sa | 0 |

(3 row(s) affected)

Viewing Permissions for Metaoutlines

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

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

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

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

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

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

2. View the results.

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

| MO_NAME | MO_OWNER | MO_ACCESS_CODE |
|-----------------|----------|----------------|
| ----- | ----- | ----- |
| TBC Metaoutline | sa | 2 |
| TBC_Archive | sa | 1 |
| TBC_Mine | sa | 0 |

(3 row(s) affected)

Troubleshooting ODBC and Connections

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

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

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

This chapter contains the following topics:

- [“Checking Common Connection Problems” on page 3-2](#)
- [“About ODBC and Connection Problems” on page 3-4](#)
- [“Isolating Connection Problems” on page 3-6](#)
- [“Correcting Connection Problems” on page 3-20](#)
- [“Using ODBC Tracing” on page 3-30](#)

Checking Common Connection Problems

If you have problems connecting to OLAP Metadata Catalog or to the external data source with Hyperion Integration Server Desktop, 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 *Hyperion Integration Server Installation Guide*.

- Are all of the following required components up and running?
 - OLAP Integration Server
 - The relational database management system (RDBMS) listener for OLAP Metadata Catalog and the data source
 - The database servers that manage OLAP Metadata Catalog and the data source databases
- Is the database client software installed and configured on the computer where OLAP 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 OLAP 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 OLAP Integration Server is installed?

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

- If necessary, is the `bin` directory for each database in the operating system path of the user who is running OLAP Integration Server?
- Does the ODBC Data Source Name (DSN) configuration identify the name of the relational database management system (RDBMS) client?
- Are OLAP Metadata Catalog and the data source configured as (ODBC) data sources on the OLAP Integration Server computer? For information on configuring ODBC data sources, see the *Hyperion Integration Server Installation Guide*.
- Are you using the ODBC data source names for OLAP Metadata Catalog and the data source databases as configured on the OLAP Integration Server computer? Do not assume that the RDBMS database name is the same as the ODBC data source name for that database.
- Are you experiencing time-out problems when connecting to the external data source?

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

- Did you create tables in OLAP Metadata Catalog? For information on creating tables for the catalog, see the *Hyperion Integration Server Installation Guide*.
- Are you having problems related to ODBC while you are creating a Hyperion Essbase outline? If so, turn on tracing, perform the action again, and analyze the results in the tracing log file. For more information, see [“Using ODBC Tracing” on page 3-30](#).

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

If none of the above steps help, review the list of database-specific problems in [“Database-Specific Problems” on page 3-4](#).

Database-Specific Problems

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

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

If none of the previous suggestions help, proceed to [“About ODBC and Connection Problems” on page 3-4](#) for an overview of connection problems, then see [“Isolating Connection Problems” on page 3-6](#).

About ODBC and Connection Problems

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

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

There are three main categories of problems that break the lines of communication between Hyperion Integration Server Desktop and the RDBMS:

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

Isolating Connection Problems

If the list of problems provided in [“Checking Common Connection Problems”](#) on [page 3-2](#) and your own knowledge fail to resolve the connection problem, the recommended strategy for diagnosing the problems is to test the components and connections starting at the RDBMS and backtrack through the connections to Hyperion Integration Server Desktop.

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

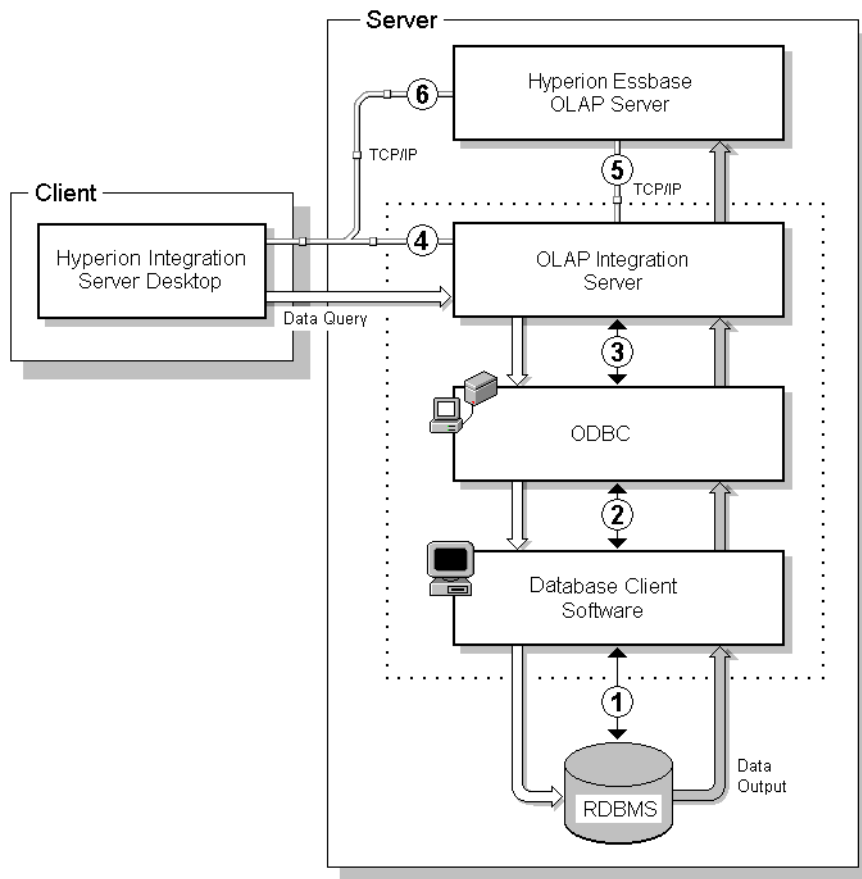


Figure 3-1: Hyperion Integration Server Components and Connections

The numbers in [Figure 3-1](#) identify the general diagnostic steps that you should take to isolate connection problems. Follow these steps as indicated, starting with the connection between the database client software and the RDBMS (1) and working backwards through the system to isolate connection problems. After you isolate a problem, refer to the appropriate components topic in [“Correcting Connection Problems” on page 3-20](#).

- To isolate connection problems, follow these steps:
 1. Attempt to connect to the RDBMS databases using a database client utility program. For more information, see [“Testing the Database Client Connection to the RDBMS” on page 3-7](#).
 2. Attempt to connect to the RDBMS databases using ODBC. For more information, see [“Testing the ODBC Connection to the RDBMS” on page 3-9](#).
 3. Attempt to connect to the RDBMS databases using OLAP Integration Server. For more information, see [“Testing the OLAP Integration Server Connection to the RDBMS” on page 3-13](#).
 4. Attempt to connect to the RDBMS databases using Hyperion Integration Server Desktop. For more information, see [“Testing the Hyperion Integration Server Desktop Connection to the RDBMS” on page 3-16](#).
 5. Attempt to connect to Hyperion Essbase OLAP Server using OLAP Integration Server. For more information, see [“Testing the OLAP Integration Server Connection to Hyperion Essbase Server” on page 3-17](#).
 6. Attempt to connect to Hyperion Essbase OLAP Server using Hyperion Integration Server Desktop. For more information, see [“Testing the OLAP Integration Server Connection to Hyperion Essbase Server” on page 3-17](#).

Testing the Database Client Connection to the RDBMS

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

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

► To test the database client connection to the RDBMS:

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

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

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

If the database client utility does not start, check to make sure the client software is installed and the required environment variables are set. For more information, see [“Correcting Database Client Problems” on page 3-24](#).

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

If the connection fails, try to connect as a user who you know has access permission, such as an administrative user. If the administrative connection succeeds, check the other user accounts and permissions in the RDBMS for the OLAP Metadata Catalog database, and carefully check user accounts with connection problems. For more information see, [“Correcting RDBMS Problems” on page 3-22](#).

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

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

If the select statement fails, check the permissions for the user name you are using to connect to the RDBMS. For more information, see [“Correcting RDBMS Problems” on page 3-22](#). Also, check that the version of the database client software you are using is compatible with the version of the RDBMS. For more information, see [“Correcting Database Client Problems” on page 3-24](#).

5. After you have successfully completed the preceding steps, try connecting to OLAP Metadata Catalog from Hyperion Integration Server Desktop.

For more information, see [“Testing the Hyperion Integration Server Desktop Connection to the RDBMS” on page 3-16](#). If the connection fails, proceed to [“Testing the ODBC Connection to the RDBMS” on page 3-9](#).

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

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

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

Testing the ODBC Connection to the RDBMS

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

On Windows NT and Windows 2000, use ODBC Administrator to test connections to the RDBMS databases. On UNIX systems, there are no standard tools for testing ODBC connections; you must manually inspect the ODBC configuration files using `ivtestlib` provided with Hyperion Integration Server, or use third-party or RDBMS-provided utilities such as `DemoODBC` or `tnsping`.

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

ODBC tracing can also be used to track down connection problems. For more information, see [“Using ODBC Tracing” on page 3-30](#).

Testing ODBC on Windows Systems

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

A data source configuration dialog box is displayed. Depending on the RDBMS and the driver you are using, you may be asked to log on to the database immediately, or you may be asked to review all the settings before testing the connection. In either case, after attempting to log on to the database, observe the resulting message boxes to determine if you connected to the RDBMS successfully.

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

For more information, see [“Correcting ODBC Problems” on page 3-25](#).

6. After you have successfully completed preceding steps, try connecting to the RDBMS from Hyperion Integration Server Desktop.

For more information, see [“Testing the Hyperion Integration Server Desktop Connection to the RDBMS” on page 3-16](#).

If the connection from Hyperion Integration Server Desktop fails, proceed to [“Testing the OLAP Integration Server Connection to the RDBMS” on page 3-13](#).

Testing ODBC on UNIX Systems

On UNIX systems, you must manually inspect the configuration files for ODBC or use an RDBMS utility to test ODBC. These configuration files include the login scripts which set the environment variables for ODBC and the `odbc.ini` file. For more information about ODBC utilities on UNIX systems, see the documentation for the RDBMS or the ODBC driver.

► To inspect the login script files:

1. Log on to the computer that runs OLAP Integration Server as the user who starts the OLAP Integration Server software.
2. In the current home directory, find the main login script file (typically, `.profile`) 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 Hyperion Integration Server script (`is.sh` or `is.csh`).

If one of these scripts is included in the main login script, make sure that the inclusion of the script file name follows the syntax for executing a script file within another script, and that the script file is specified with a complete directory path. For more information, see [“Correcting ODBC Problems” on page 3-25](#).

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 Hyperion Integration Server Desktop Connection to the RDBMS” on page 3-16](#).

If you cannot make a connection, proceed to [“Testing the OLAP Integration Server Connection to the RDBMS” on page 3-13](#).

► To inspect the `odbc.ini` file:

1. Log on to the computer that runs OLAP Integration Server as the user who starts the OLAP 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 Hyperion Integration Server script in the login script for the user. For more information, see [“Correcting OLAP Integration Server Problems” on page 3-27](#).

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 Hyperion Integration Server are listed in the ODBC Data Sources section; for example:

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

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

For Oracle:

```
[myoracle]
Driver=/export/home/users/dkendric/is200/odbc/lib/ARor815.so
Description=my oracle source

ServerName=mytns
```

For DB2:

```
[TBC_MD]
Driver=/export/home/db2inst1/sql/lib/lib/libdb2.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 /export/home/db2inst1/sqlllib/lib/libdb2.so
```

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

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

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

For more information, see [“Correcting ODBC Problems” on page 3-25](#).

Testing the OLAP Integration Server Connection to the RDBMS

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

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. Hyperion Essbase server, using TCP/IP

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

► To test the OLAP Integration Server connection to the RDBMS:

1. Verify that OLAP Integration Server is running.

If the server software is not running, start the server. If the server will not start, check that the Hyperion Integration Server environment variables are set. For more information, see [“Correcting OLAP Integration Server Problems” on page 3-27](#).

2. On the same computer that is running OLAP Integration Server, start OLAP Command Interface at a command prompt by typing:

```
olapicmd
```

If OLAP Command Interface does not start, make sure that the software is installed and that the Hyperion Integration Server environment variables are set. For more information, see [“Correcting OLAP Integration Server Problems” on page 3-27](#).

3. At the OLAP Command Interface command prompt, use the login command to connect to the OLAP Integration Server computer; for example, type:

```
login cypress
```

4. Check the release of Hyperion Integration Server by typing:

```
version
```

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

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

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

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

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

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

7. Check the status of the preceding command by typing:

```
status
```

During a member load, OLAP Command Interface connects to the data source first, then to OLAP Metadata Catalog. If the connection to the data source is successful, 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 OLAP Integration Server Problems” on page 3-27](#) and [“Correcting ODBC Problems” on page 3-25](#). Repeat steps 4 to 6 of this procedure to retest the connection. When you successfully connect to the data source, continue to the next step.

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

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

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

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

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

10. Check the status of the preceding command by typing:

status

During a member load, OLAP Command Interface connects to OLAP Metadata Catalog and then to the Hyperion Essbase server. If the connection to OLAP Metadata Catalog is successful, the message “IS Error: Unable to login to Essbase Server” is displayed. If the connection to OLAP Metadata Catalog fails, the message “IS Error: Unable to login to Catalog” is displayed.

If the connection to OLAP Metadata Catalog failed, verify the ODBC data source name, user name, and password. If the connection continues to fail, see [“Correcting OLAP Integration Server Problems” on page 3-27](#) and [“Correcting ODBC Problems” on page 3-25](#). Repeat steps 7 to 9 of this procedure to retest the connection. When you successfully connect to OLAP Metadata Catalog, continue to the next step.

11. Test the connection from Hyperion Integration Server Desktop to the RDBMS by proceeding to [“Testing the Hyperion Integration Server Desktop Connection to the RDBMS” on page 3-16](#).

Testing the Hyperion Integration Server Desktop Connection to the RDBMS

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

- To test the Hyperion Integration Server Desktop connection to the RDBMS:

1. Verify that OLAP Integration Server is running.

If the server software is not running, start the server. If the server will not start, check that the Hyperion Integration Server environment variables are set. For more information, see [“Correcting OLAP Integration Server Problems” on page 3-27](#).

2. On a computer other than the one running OLAP Integration Server, start Hyperion Integration Server Desktop.

3. Connect to the OLAP 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 OLAP Integration Server computer name. Also verify that you specified the correct user name and password for the OLAP Metadata Catalog database on the RDBMS. For more information, see [“Correcting Hyperion Integration Server Desktop Problems” on page 3-28](#).

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 OLAP Integration Server computer name. Also verify that you specified the correct user name and password for the data source database on the RDBMS. For more information, see [“Correcting Hyperion Integration Server Desktop Problems” on page 3-28](#).

Testing the OLAP Integration Server Connection to Hyperion Essbase Server

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

- To test the OLAP Integration Server connection to a Hyperion Essbase server:

1. Verify that OLAP Integration Server is running.

If the server software is not running, start the server. If the server will not start, check to ensure that the Hyperion Integration Server environment variables are set. For more information, see [“Correcting OLAP Integration Server Problems” on page 3-27](#).

2. Verify that the Hyperion Essbase server is running.

3. At a command prompt, start OLAP Command Interface by typing:

```
olapicmd
```

If OLAP Command Interface does not start, make sure that the software is installed and that the Hyperion Integration Server environment variables are set. For more information, see [“Correcting OLAP Integration Server Problems” on page 3-27](#).

4. At the OLAP Command Interface command prompt, use the login command to connect to OLAP Integration Server; for example, type:

```
login cypress
```

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

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

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

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

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

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

7. Set the Hyperion Essbase server to which you want to connect by typing:

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

8. Test the connection to the Hyperion Essbase server by attempting to perform a member load by typing:

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

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 Hyperion Essbase server is running. Verify the name of the metaoutline in Hyperion Integration Server. Also verify the name of the test application and database in Hyperion Essbase. Repeat steps 7 to 8 of this procedure to retest the connection.

After you can connect successfully to the Hyperion Essbase server, proceed to [“Testing the Hyperion Integration Server Desktop Connection to Hyperion Essbase Server” on page 3-19](#).

Testing the Hyperion Integration Server Desktop Connection to Hyperion Essbase Server

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

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

- To test the Hyperion Integration Server Desktop connection to a Hyperion Essbase server:
 1. Verify that OLAP Integration Server is running.
 2. Verify that Hyperion Essbase OLAP Server server is running.
 3. Start Hyperion Integration Server Desktop.
 4. From Hyperion Integration Server Desktop, connect to the OLAP 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 Hyperion Essbase 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 Hyperion Essbase server and the correct user name and password. For more information, see the Hyperion Essbase documentation.

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

Correcting Connection Problems

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

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

- Are all user names and passwords being used consistently across all the components and computer user accounts? For more information, see [“User Name and Password Problems” on page 3-21](#).
- Do all of the user accounts used in the system have the appropriate permissions and privileges?

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

- User account on the client computer
- User account on the server computer
- OLAP Metadata Catalog database
- Data source database
- User account for the Hyperion Essbase 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 OLAP Integration Server (for example, hyperion) configured to run all the required software?

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

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

User Name and Password Problems

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

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

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

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

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

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

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

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

Correcting RDBMS Problems

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

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

Some RDBMSs may require additional software components beyond the basic server software to enable ODBC connections. The database client software is usually required on the computer that connects to the RDBMS server using ODBC and also may be required on the RDBMS server computer.

- Do the OLAP Metadata Catalog and data source databases to which you are trying to connect exist? Have the tables for OLAP Metadata Catalog been created?

You must manually create the OLAP Metadata Catalog tables using creation scripts provided with Hyperion Integration Server. For more information, see the *Hyperion Integration Server Installation Guide*.

- What are the exact names of the OLAP Metadata Catalog and data source databases? Are the names case-sensitive?
- Does the user name that you are using to connect to the RDBMS have adequate permissions to access the database? Does the user have read and write permissions?

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

- If the RDBMS server is on a computer by itself, is there a corresponding user account on the computer for the user names you are using to connect to the databases?

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

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

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

Correcting Database Client Problems

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

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

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

- Is the database client software compatible with the RDBMS?

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

- Do the database client software utilities, such as the command-line SQL interface, start? Do they start when you are logged on as the user who runs OLAP 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 OLAP Integration Server (typically, hyperion) is configured to run the database client software.

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

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

Make sure you are using the correct names, including the exact spelling, uppercase or lowercase letters, and any required RDBMS-specific identification syntax. For more information about correcting user name and password problems, see [“Correcting RDBMS Problems” on page 3-22](#).

- 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.

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

Correcting ODBC Problems

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

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

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

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

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

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

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

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

DB2: The ODBC DSN appears as an alias when you list the database directory.

For more information about correcting user name and password problems, see [“Correcting RDBMS Problems” on page 3-22](#).

- 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?

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

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

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

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

If the preceding checks do not enable you to correct the problems connecting from ODBC to the RDBMS, try using ODBC tracing to isolate the problem. For more information, see [“Using ODBC Tracing” on page 3-30](#).

Correcting OLAP Integration Server Problems

- ▶ To correct problems connecting to the RDBMS from OLAP Integration Server, investigate the following possible causes:
 - Is OLAP Integration Server installed? Are all the required OLAP Integration Server components installed?

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

- OLAP Integration Server software.
- Related base components. (On UNIX, these components are included as part of the first item.)
- MERANT ODBC drivers (unless ODBC and drivers are already installed).

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

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

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

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

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

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

- What version of Hyperion Essbase is installed?

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

- Is port 3388 in use?
OLAP Integration Server uses port 3388 to communicate with the Hyperion Integration Server client software. If this port is being used by another program, the server software does not start.
- On UNIX, is more than one copy of `olapisvr` running? If so, all servers except one will hang indefinitely. To correct the problem, run the following command to display all programs that are currently running:

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

Many of the problems related to OLAP 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 *Hyperion Integration Server Installation Guide*.

Correcting Hyperion Integration Server Desktop Problems

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

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

- Is Hyperion Integration Server Desktop installed? Are all of the required components installed?

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

- Hyperion Integration Server Desktop software
- Related base components

- Is the Hyperion Integration Server Desktop 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 *Hyperion Integration Server Installation Guide*.

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

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

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

For more information about correcting user name and password problems, see [“Correcting RDBMS Problems” on page 3-22](#).

- Does the user account with which you are running Hyperion Integration Server Desktop have permission to connect to the computer running OLAP Integration Server?

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

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

A particular OLAP 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 OLAP Integration Server computer?

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

Using ODBC Tracing

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

Using ODBC Tracing on Windows Systems

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

► To use the tracing utility in ODBC Administrator, take the following actions:

1. Start ODBC Administrator on the computer running OLAP 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 OLAP Integration Server and Hyperion Integration Server Desktop.
4. From Hyperion Integration Server Desktop, select Connections > OLAP Metadata Catalog > Connect, and try to connect to OLAP Metadata Catalog.
5. If you can connect to OLAP Metadata Catalog, open an OLAP model and try to connect to a relational data source.
6. If either connection fails, read the `sql.log` file in the root directory.

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

Using ODBC Tracing on UNIX Systems

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

► To use ODBC tracing on UNIX systems, take the following actions:

1. On the computer running OLAP 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/odbc/lib/odbctrac.so
InstallDir=/export/home/users/hyperion/is/odbc/lib
```

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

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

Tables in OLAP Metadata Catalog

OLAP Metadata Catalog is a relational database that contains OLAP models, metaoutlines, and the information necessary to retrieve the required data from the relational data source. You can install OLAP Metadata Catalog in any supported data source. See the *Hyperion Integration Server 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” on page 4-2](#)
- [“Tables Relating to the OLAP Model” on page 4-3](#)
- [“Tables Relating to the Metaoutline” on page 4-5](#)
- [“Tables Relating to Drill-Through” on page 4-8](#)
- [“Miscellaneous Tables” on page 4-9](#)

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 a Hyperion Essbase database, Hyperion Integration Server retrieves the information stored in OLAP Metadata Catalog to determine:

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

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

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

Table 4-1: OLAP Metadata Catalog Restore Procedures

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

Tables Relating to the OLAP Model

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

- [“Tables Relating to the Metaoutline” on page 4-5](#)
- [“Tables Relating to Drill-Through” on page 4-8](#)
- [“Miscellaneous Tables” on page 4-9](#)

Table 4-2: OLAP Model Tables

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

Table 4-2: OLAP Model Tables (Continued)

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

Table 4-2: OLAP Model Tables (Continued)

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

Tables Relating to the Metaoutline

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

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

Table 4-3: Metaoutline Tables

| Table Name | Description |
|----------------|--|
| MO_FILTERS | Contains information about metaoutline filters. This information includes the filter name and the actions the filter will perform. |
| MO_ID | Contains a system-generated reference number that uniquely identifies a metaoutline within an OLAP model. |
| MO_INFO | Contains high-level information about a metaoutline: the name of the metaoutline, its description, the date it was last changed, the OLAP model on which the metaoutline is based, its owner, its access privileges, and whether the spreadsheet user of Drill-Through reports must enter the OLAP 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 Hyperion Integration Server uses to assign the properties, the numeric range that represents a range specified by the user, and Global Settings properties. 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. |
| 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. |

Table 4-3: Metaoutline Tables (Continued)

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

Tables Relating to Drill-Through

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

[Table 4-4](#) describes each table in OLAP Metadata Catalog that stores information about Drill-Through reports. For information about the other tables in OLAP Metadata Catalog, see:

- [“Tables Relating to the OLAP Model” on page 4-3](#)
- [“Tables Relating to the Metaoutline” on page 4-5](#)
- [“Miscellaneous Tables” on page 4-9](#)

Table 4-4: Drill-Through Tables

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

Miscellaneous Tables

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

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

Table 4-5: Miscellaneous Tables

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

Using OLAP Command Interface

This chapter describes the commands supported by OLAP Command Interface, the command-line tool provided with Hyperion Integration Server. The majority of these commands provide the same functionality available through Hyperion Integration Server Desktop. You use OLAP Command Interface commands and scripts to create Hyperion Essbase outlines and load Hyperion Essbase databases.

If you schedule member or data loads using Hyperion Integration Server Desktop, Hyperion Integration Server automatically creates an OLAP Command Interface 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:

- [“Starting OLAP Command Interface” on page 5-2](#)
- [“About OLAP Command Interface” on page 5-3](#)
- [“Using OLAP Command Interface Commands” on page 5-9](#)
- [“Loading Dimensions and Members Using OLAP Command Interface” on page 5-20](#)
- [“Loading Data Using OLAP Command Interface” on page 5-21](#)

For a list of all available OLAP Command Interface commands, see [“Using OLAP Command Interface Commands” on page 5-9](#).

Starting OLAP Command Interface

Before you start OLAP Command Interface, 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 OLAP Command Interface:

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

► To start OLAP Command Interface, use any of the following methods:

- From the Windows Start menu, select Programs > Hyperion Integration Server > OLAP Command Interface.
- Enter the executable name on the command line; for example, type

olapicmd

Note: If OLAP Command Interface 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 *Hyperion Integration Server Installation Guide*.

The OLAP Command Interface window opens:

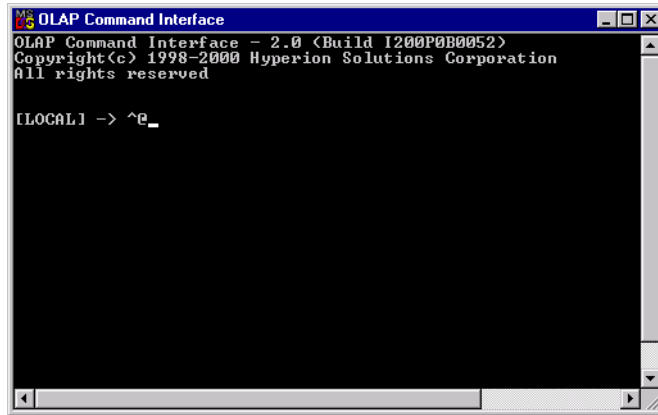


Figure 5-1: OLAP Command Interface Window

Enter commands at the command prompt. For example, type the following commands:

- Type **login localhost** to log on to a server running on the same machine.
- Type **shutdown** to stop the server.
- Type **exit** to stop OLAP Command Interface.
- Type **help** to get a list of OLAP Command Interface commands.

About OLAP Command Interface

OLAP Command Interface is a command-line tool that enables you to access Hyperion Integration Server to perform operations on a Hyperion Essbase outline and the data in a Hyperion Essbase database.

With OLAP Command Interface, 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 is difficult to enter commands correctly when using the command line. The easiest method is to use the Hyperion Integration Server Desktop OLAP Metaoutline standard user interface to perform a data or member load, which automatically creates a script file. You can then edit the script file as needed. Run all the commands in a script file by redirecting the standard input to use your command file. For example, `olapicmd mycmd.cbs`, or see [“Running OLAP Command Script Files” on page 5-8](#).

Using SET and LOAD Commands

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

Use SET commands as follows:

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

The STATUS command (see [“STATUS” on page 5-10](#)) does not check or validate the syntax for SET commands.

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

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

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

See [“Loading Dimensions and Members Using OLAP Command Interface” on page 5-20](#) for a sample sequence of commands for loading members.

See [“Loading Data Using OLAP Command Interface” on page 5-21](#) for a sample sequence of commands for loading data.

Using Informational Commands

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

Using Command Syntax and Execution

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

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

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

You must separate the statements with a semicolon. The final semicolon is optional. Do not use semicolons as part of data source names, because OLAP Command Interface 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 *Hyperion Integration Server 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 Hyperion Integration Server Desktop to schedule member or data loads. An OLAP Command Interface batch file and a corresponding .cbs file are created automatically in the *ISHOME\Batch* directory. You can reference, modify, and reuse the .cbs file for future member and data loads. Be sure to check the **Save Script Only** option instead of **Schedule**. For more information, see [“Recording Member and Data Load Scripts” on page 5-6](#).

Recording Member and Data Load Scripts

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

- To record a load script:
1. Start Hyperion Integration Server Desktop 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 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 OLAP Command Interface 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 OLAP Command Interface commands. You can run a script file from the operating system command line or from within an operating system batch file.
- On Windows NT and Windows 2000, a batch file (.bat extension) is an operating system file that calls multiple OLAP Command Interface scripts and is used to run multiple sessions of OLAP Command Interface 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, OLAP Command Interface 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 OLAP Command Interface:

```
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 Hyperion Essbase from the server console and generates outlines for a sample database. In the following example, the STATUS command (see [“STATUS” on page 5-10](#)) is used to check the success or failure of each command. The STATUS command returns the execution status in the OLAP Command Interface window.

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

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

```
olapicmd -folap_tbc.cbs
```

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

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

Using OLAP Command Interface Commands

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

The following commands are described in the subtopics:

- [“LOGIN” on page 5-10](#)
- [“STATUS” on page 5-10](#)
- [“VERSION” on page 5-11](#)
- [“SETSOURCE” on page 5-11](#)
- [“SETTARGET” on page 5-12](#)
- [“SETCATALOG” on page 5-12](#)
- [“LOADMEMBER” on page 5-13](#)
- [“LOADDATA” on page 5-15](#)
- [“LOADALL” on page 5-17](#)
- [“LOGOUT” on page 5-19](#)
- [“SHUTDOWN” on page 5-19](#)
- [“EXIT” on page 5-20](#)

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

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

LOGIN

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

Syntax

LOGIN *Ishost*

| | |
|---------------|--|
| <i>Ishost</i> | TCP/IP name or number of the computer where OLAP Integration Server is running |
|---------------|--|

Example

LOGIN cypress

STATUS

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

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

Syntax

STATUS

Example

Typing STATUS after issuing the LOADMEMBER command returns the following message in the OLAP Command Interface window:

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

VERSION

The VERSION command returns the Release number of the Hyperion Integration Server software you are running.

Syntax

VERSION

Example

VERSION returns the release number for Hyperion Integration Server software.

SETSOURCE

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

Syntax

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

| | |
|-----------------------|--|
| <i>ODBCdatasource</i> | The ODBC data source name configured on the computer where OLAP Integration Server runs—case-sensitive |
| <i>username</i> | The name for logging on to the ODBC data source—case-sensitive |
| <i>password</i> | The password for logging on to the ODBC data source—case-sensitive |

Example

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

SETTARGET

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

Syntax

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

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

Example

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

SETCATALOG

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

Syntax

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

| | |
|-----------------------|--|
| <i>ODBCdatasource</i> | The ODBC data source name configured on the computer where OLAP Integration Server runs—case-sensitive |
| <i>username</i> | The name for logging on to the ODBC data source—case-sensitive |
| <i>password</i> | The password for logging on to the ODBC data source—case-sensitive |

Example

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

LOADMEMBER

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

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

Syntax

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

| | |
|------------------------------------|--|
| <i>Metaoutline</i> | The name of the metaoutline—case-sensitive. This metaoutline is used to extract data and create a Hyperion Essbase outline. |
| <i>EssbaseApplication</i> | The name of the Hyperion Essbase application on the target Hyperion Essbase OLAP Server computer where the Hyperion Essbase outline is created. |
| <i>EssbaseDatabase</i> | The name of the Hyperion Essbase database on the target Hyperion Essbase OLAP Server computer. OLAP Integration Server applies this name to the Hyperion Essbase outline. |
| <i>Filtername/Filter_ID_Number</i> | Optional. You can specify a filter name or filter ID to use when loading data. Because the filter name and filter ID are the same, do not specify both. If you do not specify either, the default name (*DEFAULT) is used. The default filter ID is 1; any additional filters are numbered sequentially, starting with the number 2. If the filter name contains a space, enclose it in double quotation marks. For information about creating filters, see the <i>Hyperion Integration Server Desktop OLAP Metaoutline User's Guide</i> . |

| | |
|---------------------------------|--|
| <i>EsscmdScriptname</i> | Optional. The name of an ESSCMD script. The script that you define instructs Hyperion Essbase to perform specified actions on the Hyperion Essbase database. These actions might include changing a member tag or loading data from a worksheet into user-defined members after you load the data from the relational data source. The ESSCMD script must be in the <i>ISHOME/essscript</i> directory. |
| <i>Delete all members first</i> | Optional. Specify Y or N. Enter Y (Yes) for OLAP Integration Server to remove all members from the existing database and then recreate the database by using the members in the metaoutline. This process is slower than creating or updating a Hyperion Essbase outline without deleting the members; therefore, do not delete all members first unless it is necessary. You should delete all members if, for example, you know that some members have been removed from the OLAP metaoutline and you want to build a Hyperion Essbase outline containing a smaller set of members. The default is N (No). |
| <i>EssbaseUsers</i> | Optional. Lists all user IDs that can access the Hyperion Essbase application and database into which you are loading the members. Commas separate user IDs. For example, @@USERS="admin", "ljones", "ksmith". |

Example

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

LOADDATA

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

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

Syntax

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

| | |
|------------------------------------|--|
| <i>Metaoutline</i> | The name of the metaoutline. OLAP Integration Server uses the specified metaoutline to extract data from the relational data source to create a Hyperion Essbase outline. |
| <i>EssbaseApplication</i> | The name of the Hyperion Essbase application on the target Hyperion Essbase OLAP Server computer where the Hyperion Essbase outline is created. |
| <i>EssbaseDatabase</i> | The name of the Hyperion Essbase database on the target Hyperion Essbase OLAP Server computer. OLAP Integration Server applies this name to the Hyperion Essbase outline. |
| <i>Filtername/Filter_ID_Number</i> | Optional. You can specify a filter name or filter ID to use when loading data. Because filter name and filter ID are the same, do not specify both. If you do not specify either, the default name (*DEFAULT) is used. The default filter ID is 1; any additional filters are numbered sequentially, starting with the number 2. If the filter name contains a space, enclose it in double quotation marks. For information about creating filters, see the <i>Hyperion Integration Server Desktop OLAP Metaoutline User's Guide</i> . |

| | |
|-------------------------|--|
| <i>CalcScriptname</i> | Optional. The name of the calc script. The calc script determines how Hyperion Essbase calculates the data values in the database. If you do not select a calc script, OLAP Integration Server does not perform a calculation. |
| <i>EsscmdScriptname</i> | Optional. The name of an ESSCMD script. The script that you define instructs Hyperion Essbase to perform specified actions on the Hyperion Essbase database. These actions might include changing a member tag or loading data from a worksheet into user-defined members after you load the data from the relational data source. The ESSCMD script must be in the <i>ISHOME/essscript</i> directory. |
| <i>EssbaseUsers</i> | Optional. Lists all user IDs that can access the Hyperion Essbase application and database into which you are loading the members. Commas separate user IDs. For example, @@USERS="admin", "ljones", "ksmith". |

Example

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

LOADALL

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

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

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

Syntax

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

| | |
|---------------------------|---|
| <i>Metaoutline</i> | The name of the metaoutline. OLAP Integration Server uses the specified metaoutline to extract data from the relational data source to create a Hyperion Essbase outline. |
| <i>EssbaseApplication</i> | The name of the Hyperion Essbase application on the target Hyperion Essbase OLAP Server computer where the Hyperion Essbase outline is created. |
| <i>EssbaseDatabase</i> | The name of the Hyperion Essbase database on the target Hyperion Essbase OLAP Server computer. OLAP Integration Server applies this name to the Hyperion Essbase outline. |

| | |
|------------------------------------|---|
| <i>Filtername/Filter_ID_Number</i> | Optional. You can specify a filter name or filter ID to use when loading data. Because filter name and filter ID are the same, do not specify both. If you do not specify either, the default name (*DEFAULT) is used. The default filter ID is 1; any additional filters are numbered sequentially, starting with the number 2. If the filter name contains a space, enclose it in double quotation marks. For information about creating filters, see the <i>Hyperion Integration Server Desktop OLAP Metaoutline User's Guide</i> . |
| <i>CalcScriptname</i> | Optional. The name of the calc script. The calc script determines how Hyperion Essbase calculates the data values in the database. If you do not select a calc script, OLAP Integration Server does not perform a calculation. |
| <i>EsscmdScriptname</i> | Optional. The name of an ESSCMD script. The script that you define instructs Essbase to perform specified actions on the Hyperion Essbase database. These actions might include changing a member tag or loading data from a worksheet into user-defined members after you load the data from the relational data source. The ESSCMD script must be in the <code>hyperion\is\esscript</code> directory. |
| <i>Delete all members first</i> | Optional. Specify Y or N. Enter Y (Yes) for Hyperion Integration Server to remove all members from the existing database and then recreate the database by using the members in the metaoutline. This process is slower than creating or updating a Hyperion Essbase outline without deleting members; therefore do not delete all members unless it is necessary. You should delete all members if, for example, you know that some members have been removed from the OLAP metaoutline and you want to build the Hyperion Essbase outline containing a smaller set of members. The default is N (No). |

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

Example

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

LOGOUT

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

Syntax

```
LOGOUT
```

Example

LOGOUT logs you out of OLAP Integration Server.

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

SHUTDOWN

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

Syntax

```
SHUTDOWN
```

Example

SHUTDOWN shuts down OLAP Integration Server.

EXIT

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

Syntax

EXIT

Example

EXIT logs you out of OLAP Integration Server and closes the OLAP Command Interface window.

Loading Dimensions and Members Using OLAP Command Interface

You can perform a member load by using OLAP Command Interface instead of Hyperion Integration Server Desktop.

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

- ▶ To load members and dimensions when using OLAP Command Interface, follow the same steps for loading members as you would when using Hyperion Integration Server Desktop:

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

```
LOGIN cypress
```

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

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

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

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

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

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

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

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

Loading Data Using OLAP Command Interface

You can perform a data load by using OLAP Command Interface instead of Hyperion Integration Server Desktop.

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

- To load data by using OLAP Command Interface, follow the same steps for loading data as you would using Hyperion Integration Server Desktop:

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

```
LOGIN FIG
```

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

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

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

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

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

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

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

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


OLAP Integration Server Error Messages

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

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

Member Load Error Messages

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

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

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

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

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

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

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

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

| Message Number | Message Content |
|----------------|--|
| 357 | Essbase Error: <i>error definition</i> . |
| 358 | Failed to get user attributes. |
| 360 | Failed to add member to outline. |
| 361 | Failed to calculate the data. |
| 362 | Failed to get parent. |
| 363 | Failed to get member. |
| 364 | Failed to move member. |
| 365 | Failed to get shared member. |
| 366 | Failed to get member information. |
| 367 | Failed to sort outline. |
| 377 | IS Error: Unable to delete outline. |
| 378 | IS Error: SQL Generation for member load failed. |
| 379 | IS Error: Database close operation failed after load. |
| 380 | IS Error: Catalog close operation failed after load. |
| 381 | IS Error: Essbase close operation failed after load. |
| 382 | IS Error: No OLAP model name specified. |
| 383 | IS Error: Failed to open OLAP model <i>OLAP model</i> . |
| 384 | IS Error: Failed to close OLAP Metadata Catalog <i>OLAP Metadata Catalog</i> . |
| 385 | Failed to create virtual model. |
| 386 | Member load completed with errors. |
| 387 | Member load terminated due to error. |

Data Load Error Messages

Table A-2 lists the error messages that OLAP Integration Server may generate during a data load. Italicized words represent variable names inserted in the error message. Refer to the error message number in the first column and make a note of it if you need to contact Hyperion Technical Support with a problem.

Table A-2: OLAP Integration Server Data Load Error Messages

| Message Number | Message Content |
|----------------|--|
| 034 | Cell load exceptions encountered. |
| 047 | Data load exceptions encountered. <i>?</i> , <i>amount</i> . |
| 063 | Failed to activate <i>Essbase_application.Essbase.database</i> . |
| 082 | Failed to execute esscmd script <i>esscmd script</i> . |
| 388 | IS Error: Failed to get dataload SQL. |
| 389 | IS Error: Failed to get dimbuild SQL. |
| 390 | IS Error: No. of dataload SQLs not equal to no of hierarchies. |

Drill-Through Report Error Messages

Drill-Through reports are created using the Hyperion Integration Server OLAP Desktop OLAP Metaoutline standard user interface and are viewed using Hyperion Essbase Spreadsheet Add-in for Excel or Lotus 1-2-3.

[Table A-3](#) lists the error messages that OLAP Integration Server may generate to Hyperion Essbase users viewing Drill-Through reports. Italicized words represent variable names inserted in the error message. Refer to the error message number in the first column and make a note of it if you need to contact Hyperion Technical Support with a problem.

Table A-3: OLAP Integration Server Drill-Through Report Error Messages

| Message Number | Message Content |
|-----------------------|--|
| 081 | Failed to enable DTS Member <i>member</i> . |
| 087 | Failed to get the universal member handle. |
| 125 | Intersection element <i>member</i> does not exist in <i>table.column</i> . |
| 131 | Invalid parameter count for the drill-through request. Aborting... |
| 153 | Received NULL pointer in DTAttributes() for Report = <i>Drill-Through report</i> . |
| 154 | Received NULL pointer in GetDTData () for Report = <i>Drill-Through report</i> . |
| 155 | Received NULL pointer in GetDTDomain (). |
| 156 | Received NULL pointer in GetDTReport (). |
| 157 | Received NULL pointer in SetDTReport (). |
| 183 | Too many Drill-Through reports defined (exceeded Essbase metadata size limit). Drill-Through is disabled for this outline. |
| 186 | Unknown exception in GetDTReport (). |
| 359 | Essbase Error: Invalid Drill-Through Metadata. |
| 369 | IS Error: Data load completed with errors. |
| 370 | IS Error: Data load terminated due to errors. |
| 371 | IS Error: Unable to log in to data source. |

Table A-3: OLAP Integration Server Drill-Through Report Error Messages (Continued)

| Message Number | Message Content |
|-----------------------|--|
| 372 | IS Error: Unable to log in to OLAP Metadata Catalog. |
| 373 | IS Error: Unable to log in to Essbase Server. |
| 374 | IS Error: Unable to read Metaoutline information. |
| 375 | IS Error: Data calculation failed. |
| 376 | IS Error: Esscmd script execution failed. |

Miscellaneous Error Messages

[Table A-4](#) lists the miscellaneous error messages that OLAP Integration Server may generate during OLAP model and metaoutline creation and during data and member loads. Italicized words represent variable names inserted in the error message. Refer to the error message number in the first column and make a note of it if you need to contact Hyperion Technical Support with a problem.

Table A-4: OLAP Integration Server Miscellaneous Error Messages

| Message Number | Message Contents |
|-----------------------|---|
| 004 | <i>Metaoutline</i> failed validation. Database measures not specified. |
| 007 | <i>Add job</i> failed. |
| 017 | <i>Syntax</i> Syntax error at <i>character location</i> in the filter expression <i>filter</i> . |
| 018 | <i>Member</i> cannot have aggregation filter. Only leaf members can have aggregation filters. |
| 054 | Error message unavailable for this error. |
| 057 | Essbase server on <i>Essbase server computer</i> is not enabled with Hyperion Integration Server option. Please inform your system administrator. |
| 084 | Failed to get job info (OS error <i>Job-ID</i>). |

Table A-4: OLAP Integration Server Miscellaneous Error Messages (Continued)

| Message Number | Message Contents |
|----------------|---|
| 086 | Failed to get the ODBC message. |
| 090 | Failed to remove <i>temporary batch file</i> (OS error <i>operating system error</i>). |
| 091 | Failed to remove job (OS error <i>operating system error</i>). |
| 092 | Failed to remove old job (OS error <i>operating system error</i>). |
| 094 | Failed to schedule job (OS error <i>operating system error</i>). |
| 096 | Fatal error. |
| 124 | Internal system error. Please contact Technical Support with the error number 1999999. |
| 127 | Invalid command line option <i>OLAP Integration Server switch</i> . |
| 128 | Invalid datatype specification <i>data type</i> . |
| 133 | Invalid session identifier; please log in to Hyperion Integration Server again. |
| 159 | Replace Failed <i>member transformation</i> . |
| 185 | Trying to access invalid memory. Contact Technical Support. |
| 187 | Unknown ODBC exception encountered while closing database. |
| 206 | <i>Syntax</i> Syntax error at <i>character location</i> in the filter expression <i>filter</i> . |
| 207 | Hyperion Integration Server is already running on this machine. |
| 208 | Failed to release locks. |
| 333 | ODBC Error: Encountered unknown ODBC exception while opening database. |
| 334 | ODBC Error: Encountered unknown ODBC exception while closing database. |
| 336 | ODBC Error: Failed to get ODBC Error message for error number. |
| 338 | IS Error: Error during parsing Load Parameters. |
| 343 | Essbase Error: Essbase Server on <i>Essbase server computer</i> is not enabled with Hyperion Integration Server option. |

Table A-4: OLAP Integration Server Miscellaneous Error Messages (Continued)

| Message Number | Message Contents |
|----------------|---|
| 344 | Failed to unload database. |
| 348 | Failed to get information for Database <i>database</i> . |
| 391 | IS Error: Unable to get tables for pattern name <i>SQL regular expression pattern</i> from data source. |
| 392 | IS Error: Unable to get columns for table <i>relational table</i> from data source. |
| 393 | IS Error: Unable to extract foreign keys for tables <i>relational table</i> , <i>relational table</i> from data source. |
| 394 | IS Error: Failed to bind columns. |
| 395 | IS Error: Failed to fetch next row. |
| 396 | IS Error: Failed to get number of columns. |
| 397 | IS Error: Failed to get column attributes. |
| 398 | IS Error: Failed to get statement handle. |
| 399 | IS Error: Failed to release statement handle. |
| 400 | IS Error: Failed to get number of rows. |
| 401 | IS Error: Failed to connect to Hyperion Integration Server. |

#MISSING. *See* [missing data \(#MISSING\)](#).

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

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

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

aggregate. *See* [consolidate](#).

aggregation level. *See* [consolidation level](#).

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

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

alternate name. *See* [alias](#).

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

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

ARBORPATH. An environment variable that specifies the Hyperion Essbase root directory.

Architect. *See* [Hyperion Integration Server Desktop](#).

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

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

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

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

Builder. See [Hyperion Integration Server Desktop](#).

calc script. See [calculation script](#).

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

Catalog. See [OLAP Metadata Catalog](#).

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

child. A member that has a parent above it in the database outline. A child may have peers (siblings) that exist at the same layer of the database outline.

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 [OLAP Command Interface](#).

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. After you enter or load data into child entities, you perform a consolidation to aggregate the data through the organization. As data consolidates, intercompany processing, conversion methods, equity adjustments, and minority ownerships perform calculations on the data.

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

data block. *See* [block](#).

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

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

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

data load rules. A set of criteria or rules that Hyperion Essbase uses to determine how to load data from a text-based file or a relational data set into a Hyperion Essbase database.

data point. *See* [cell](#).

data source. External data, such as a text file, spreadsheet file, or SQL database that will be loaded into a Hyperion Essbase 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. An individual who administers database servers, such as Hyperion Essbase, and who may also design, maintain, and create databases.

database outline. *See* [outline](#).

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

DBMS. See [relational database management system \(RDBMS, DBMS\)](#).

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

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

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 Hyperion Essbase dimension in a Hyperion Integration Server metaoutline.

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

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

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

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

Dynamic Calc And Store members. A member that the Hyperion Essbase server calculates only upon the first retrieval of the value. The server then stores the calculated value in the database. Subsequent retrievals of a Dynamic Calc And Store member do not require calculating.

Dynamic Calc members. A member that the Hyperion Essbase server calculates only at retrieval time. The server discards calculated values after the retrieval request is complete.

dynamic calculation. In Hyperion Essbase, a calculation that occurs only when you retrieve data on a member that has been tagged as Dynamic Calc or Dynamic Calc And Store. The values for that member are calculated at retrieval time instead of being precalculated during batch calculation.

Dynamic Time Series. A process that is used to perform dynamic period-to-date reporting.

Dynamic Time Series members. Predefined members that are used to perform Dynamic Time Series reporting.

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

Essbase OLAP server. A database server that locates and accesses multidimensional data.

Essbase outline. *See* [outline](#).

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

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

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

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

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

formula. In Hyperion Essbase, a combination of operators and functions as well as dimension names, member names, and numeric constants. Formulas are used to calculate relationships between members of a Hyperion Essbase database.

@VAR(Actual, Budget) is an example of a formula.

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

generation name. A unique name that describes a generation.

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

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

Hyperion Integration Server Desktop. The client component of the Hyperion Integration Server product family. This tool is used to create OLAP models and metaoutlines, and to load data into Hyperion Essbase databases.

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

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

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

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

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

input block. A type of data block that contains 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* [OLAP Integration Server](#).

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

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

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

join columns. In Hyperion Integration Server, 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 Hyperion Integration Server Desktop 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 your source relational database.

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

level 0 block. A data block that is created for sparse member combinations when all 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 Hyperion Integration Server, 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, Hyperion Integration Server is capable of loading members and data into the Hyperion Essbase database.

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

logical column. In Hyperion Integration Server, a column created by manipulating the data in one or more physical columns. *See also* [column](#). *Contrast with* [physical column](#).

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

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

MDDB. *See* [multidimensional database \(MDDB\)](#).

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

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

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

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

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

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

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

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

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

Move mode. In Hyperion Integration Server, 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 type of database that enables referencing data through three or more dimensions. An individual record is the intersection of a point for a set of dimensions. *Contrast with* [relational database](#).

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

numeric transformation. In Hyperion Integration Server, 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 \(ODBC\)](#).

OLAP. *See* [online analytical processing \(OLAP\)](#).

OLAP Architect. *See* [Hyperion Integration Server Desktop](#).

OLAP Builder. *See* [Hyperion Integration Server Desktop](#).

OLAP Catalog. *See* [OLAP Metadata Catalog](#).

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

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

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

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

OLTP. *See* [online transaction processing \(OLTP\)](#).

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

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

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

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

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

pass-through transformations. In Hyperion Integration Server, a feature that enables you to use functions specific to your relational database management system (RDBMS) to extract data values for columns. You can provide a statement that Hyperion 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. In Hyperion Integration Server, a parameter that you set to control the time and size of the queries made to the data source.

RDBMS. See [relational database management system \(RDBMS, DBMS\)](#).

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

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

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

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

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

right frame. In the Hyperion Integration Server Desktop 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.

roll-up. See [consolidate](#).

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

server. See [Essbase OLAP server](#) and [OLAP Integration Server](#).

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

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

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

SQL. See [structured query language \(SQL\)](#).

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

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

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

storage manager. See [Hyperion Essbase kernel](#).

string. A sequence of characters treated as a unit.

structured query language (SQL). A computer language used to access data in relational databases.

supervisor. A defined type of user who has full access to all applications, databases, related files, and security mechanisms for a server.

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

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

time dimension. A dimension type that defines how often data is collected and updated, such as per fiscal or calendar periods. Only one dimension may be tagged as time; a time dimension is not required.

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

two-pass calculation. A Hyperion Essbase property that is used to recalculate members that are dependent on the calculated values of other members. Two-pass members are calculated during a second pass through the database outline.

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

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

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

upper-level block. A type of data block that is created for sparse member combinations when at least one of the sparse members is a parent-level member.

user-defined attribute. See [UDA](#).

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

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

validation. (1) In Hyperion Essbase, a process of checking a rules file against the outline to make sure that the rules file is valid. (2) In Hyperion Integration Server, 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.

Index

A

access

- codes, for metaoutlines, [2-9](#)
- codes, for OLAP models, [2-8](#)
- exclusive access mode, [2-3](#)
- multi-user, [2-2](#)
- privileges, common problems, [3-2](#)
- problems, correcting, [3-20](#)
- setting, [2-6](#)
- standard access mode, [2-2](#)

accessing

- data sources, [3-2](#)
- servers and databases, troubleshooting, [3-1](#)

accounts

- problems in the RDBMS, [3-8](#)
- user, administrative, [3-8](#)
- user, problems, [3-21](#)

administrative accounts, using, [3-8](#)

Architect. *See* Hyperion Integration Server Desktop

AT service, [5-20](#), [5-21](#)

audience, skills required, [vii](#)

B

backtracking connection problems, [3-6](#)

backups, OLAP Metadata Catalog, [4-2](#)

bat extension files, [5-7](#)

batch files

- OLAP Command Interface and, [5-3](#)
- running, [5-8](#)
- running OLAP Command Interface scripts with, [5-7](#)
- scheduling manually, [5-20](#), [5-21](#)

batch mode, OLAP Command Interface, [5-3](#)

binding, requirement for DB2 databases, [3-25](#)

brackets, in OLAP Command interface syntax, [5-6](#)

buffer size, setting for OLAP Integration Server, [1-5](#)
Builder. *See* Hyperion Integration Server Desktop

C

catalog. *See* OLAP Metadata Catalog

CB_CONCURRENCY table

- column descriptions, [2-4](#)
- defined, [4-9](#)

CB_OBJECT_TYPE column, data defined, [2-4](#)

CB_VERSION table, [4-9](#)

cbs extension files, [5-7](#)

checking

- common connection problems, [3-2](#)
- release number of Hyperion Integration Server, [3-14](#)
- script syntax for loads, [5-6](#)

client software

- port number, [3-28](#)
- problems, correcting, [3-28](#)
- RDBMS, common problems, [3-2](#)
- RDBMS, correcting problems, [3-24](#)
- troubleshooting, [3-1](#)

codes, access

- for metaoutlines, [2-9](#)
- for OLAP models, [2-8](#)

command line

- loading data from, [5-21](#)
- loading members from, [5-20](#)
- OLAP Command Interface, [5-2](#), [5-3](#)

- command syntax, OLAP Command Interface, 5-6
- command-line interface, using, 5-1
- commands, OLAP Command Interface, 5-9
- commands, status of, 5-10
- common problems
 - areas of failure, 3-4
 - database-specific, 3-4
 - with connections, 3-2
- communication problems, 3-5
- compatibility
 - between RDBMS and RDBMS client, 3-24
 - ODBC drivers, 3-25
 - problems with product versions, 3-14
- components, software, common problems, 3-2, 3-5
- concurrent users, 2-4
- configuration problems, 3-5
- connecting
 - database client to RDBMS, 3-7
 - problems with data source, 3-2, 3-4
 - problems with OLAP Metadata Catalog, 3-2, 3-4
- connections
 - areas of failure, 3-4
 - common problems, 3-2
 - isolating problems, 3-6
 - LOADMEMBER command, 3-13
 - OLAP Metadata Catalog, 3-2, 3-4
 - path through components, 3-6
 - problems, correcting, 3-20
 - problems, diagnosing, 3-4
 - problems, isolating, 3-6
 - testing, 3-9, 3-16
 - troubleshooting, 3-1
- consulting services, xii
- conventions, document, x
- correcting
 - connection problems, 3-1, 3-20
 - data source problems, 3-25
 - database client problems, 3-24
 - Hyperion Integration Server Desktop problems, 3-28
 - ODBC problems, 3-1, 3-25
 - OLAP Integration Server problems, 3-27
 - password problems, 3-21
 - RDBMS problems, 3-22
 - user name problems, 3-21

- creating
 - Hyperion Essbase applications, 2-2
 - Hyperion Essbase databases, 2-2
 - OLAP Integration Server users, 2-2
- critical server error messages, 1-4

D

- data loads
 - checking script syntax, 5-6
 - command line and, 5-21
 - incremental, script syntax, 5-9
 - LOADALL command, 5-17
 - LOADDATA command, 5-15
 - OLAP Command Interface and, 5-20, 5-21
 - OLAP Metadata Catalog and, 1-1, 4-2
 - optimizing, 1-5
 - scripts, recording, 5-6
 - time required, 5-5
- data sources
 - accessing, 3-2
 - connecting to, 5-20, 5-21
 - connection problems, 3-2, 3-4
 - connections to, diagram, 3-6
 - identifying before loading, 5-11
 - problems, correcting, 3-25
- data, retrieving, 1-1
- database client software
 - common problems, 3-2
 - environment problems, 3-7, 3-24
 - problems, correcting, 3-24
 - testing connection to RDBMS, 3-7
- database servers
 - common problems, 3-2
 - problems, correcting, 3-22
- databases
 - connections to, diagram, 3-6
 - vendor-specific problems, 3-4
- DB2
 - common problems, 3-4
 - database binding requirement, 3-25
- debug messages, 1-4

- deleting
 - locks, [2-4](#), [2-6](#)
 - OLAP Integration Server users, [2-2](#)
 - read and write locks, [2-6](#)
 - users, [2-2](#)
- DemoODBC program, [3-9](#)
- diagnosing problems (connections, ODBC), [3-1](#)
- diagram, connections path, [3-6](#)
- disconnecting, users, [2-2](#)
- DNS, problems, [3-5](#)
- document conventions, [x](#)
- documentation, list of, [viii](#)
- domain name server, problems, [3-5](#)
- Drill-Through reports, tables relating to, [4-8](#)
- driver file, ODBC, specifying on UNIX, [3-13](#)
- drivers, ODBC, problems, [3-25](#)
- DTR_ATTRIBUTES table, [4-8](#)
- DTR_INFO table, [4-8](#)

E

- environment variables
 - common problems, [3-2](#)
 - for database client, problems, [3-7](#), [3-24](#)
 - ODBC problems, [3-9](#)
- error messages, setting logging level, [1-4](#)
- Essbase
 - server, connecting to, [5-20](#)
- Essbase. *See* Hyperion Essbase
- essbase.mdb file, [1-4](#), [1-6](#)
- exclusive access mode, [2-3](#)
- executable name
 - for OLAP Command Interface, [5-2](#)
 - for OLAP Integration Server, [1-2](#)
- EXIT command, [5-20](#)
- extra locks
 - deleting, [2-6](#)
 - viewing, [2-4](#)

F

- file name, Hyperion Essbase messages file, [1-4](#)
- file names, [1-4](#)

- files
 - batch. *See* batch files
 - log. *See* log files
 - script. *See* scripts
- fixing. *See* troubleshooting

H

- host names, problems, [3-20](#)
- Hyperion Essbase
 - message file, [1-6](#)
 - relationship to OLAP Integration Server, [1-1](#)
 - server name problems, [3-20](#)
 - server, connecting to, [5-21](#)
 - server, message database, [1-4](#)
 - target data server, specifying, [5-12](#)
 - testing connection from Hyperion Integration Server Desktop, [3-19](#)
 - testing connection from OLAP Integration Server, [3-17](#)
 - users, defined, [2-1](#)
- Hyperion Integration Server
 - connection path, diagram, [3-6](#)
 - release number, checking, [3-14](#)
- Hyperion Integration Server Desktop
 - common problems, [3-2](#)
 - port number, [1-5](#), [3-28](#)
 - problems, correcting, [3-28](#)
 - testing connection to Hyperion Essbase, [3-19](#)
 - testing connection to RDBMS, [3-16](#)

I

- incompatibility, problems with product versions, [3-14](#)
- incremental loads, script syntax, [5-9](#)
- informational messages, [1-4](#)
- Integration Server. *See* OLAP Integration Server
- interactive mode, OLAP Command Interface, [5-4](#)
- INTERSOLV. *See* MERANT, ODBC driver
 - problems, correcting
- introduction, [vii](#)
- is.csh file, using, [3-11](#)
- is.sh file, using, [3-11](#)

isolating, connection problems, overview, 3-6
ivtestlib program, 3-9

J

JOIN_HINTS table, 4-5

L

levels of access permissions, 2-8
library path, common problems, 3-3
listeners, setting, 1-5
LOAD commands, 5-4, 5-5
LOADALL command, described, 5-17
LOADDATA command
 described, 5-15
 example of, 5-21
loading
 data, optimizing, 1-5
 data, time required, 5-5
 from the command line, 5-21
 OLAP Command Interface and, 5-20, 5-21
 OLAP Metadata Catalog and, 1-1, 4-2
 test, members into Hyperion Essbase, 3-17
LOADMEMBER command
 connections made by, 3-13
 described, 5-13
 example of, 5-21
 using to test connections, 3-13
locks
 defined, 2-2
 deleting, 2-4, 2-6
 metaoutlines, 2-2
 OLAP models, 2-2
 optimistic, 2-2
 read, deleting, 2-6
 read, viewing, 2-5
 storage location in the OLAP Metadata Catalog,
 2-4
 timestamps and, 2-4
 unreleased, 2-4
 viewing, 2-4
 working with, 2-1, 2-2
 write, deleting, 2-6
 write, viewing, 2-5

log files
 changing name and location, 1-4
 contents, 1-6
 messages recording level, 1-4
 OLAP Integration Server, viewing, 1-6
logging level, 1-4
LOGIN command
 described, 5-10
 example of, 5-20, 5-21
login problems, correcting, 3-21
login script
 diagnosing problems, 3-11
 problems, 3-7
LOGOUT command, 5-19

M

managing
 locks and permissions, 2-1
 user names and passwords, 3-21
MB_FILTER table, 4-3
MB_TX_RULE table, 4-3
member loads
 checking script syntax, 5-6
 command line, 5-20, 5-21
 LOADMEMBER command, 5-13
 OLAP Metadata Catalog and, 1-1, 4-2
 scripts, recording, 5-6
MERANT, ODBC driver problems, correcting, 3-25
message file
 Hyperion Essbase, 1-6
messages, debug, 1-4
metaoutlines
 access permissions, 2-6
 locks, 2-2
 opening in exclusive access mode, 2-3
 opening in standard access mode, 2-2
 overwriting, 2-3
 permissions, viewing, 2-9
 restoring, 4-2
 tables in OLAP Metadata Catalog, 4-5
 users, 2-4
MO_ACCESS_CODE column, data defined, 2-9
MO_DESCRIPTIONS table, 4-7
MO_FILTERS table, 4-6
MO_INFO table, 2-9, 4-6

MO_INTERSECTION table, 4-8
 MO_PROPERTIES table, 4-6
 MOD_ATTRIBUTES table, 4-6
 MOD_INFO table, 4-6
 Model. *See* Hyperion Integration Server Desktop
 MODEL_ACCESS_CODE column
 data defined, 2-8
 MS_INFO table, 4-7
 MS_PREDICATES table, 4-7
 MS_TX_RULE table, 4-7
 MSR_INFO table, 4-7
 MSR_PREDICATES table, 4-7
 MSR_TX_RULE table, 4-7
 multi-threaded server, 1-1
 multi-user access, 2-2

N

names, server problems, 3-20
 network listeners, setting, 1-5
 network problems, 3-5
 non-critical error messages, 1-4

O

OA_INFO table, 4-3
 OD_DETAILS table, 4-3
 OD_INFO table, 4-4
 ODBC
 common problems, 3-3
 driver compatibility, 3-25
 driver file, specifying on UNIX, 3-13
 environment problems, 3-7
 problems, correcting, 3-25
 testing connection to RDBMS, 3-9
 tracing utility, using, 3-30
 troubleshooting, 3-1
 ODBC Administrator, 3-9
 ODBC Data Sources, odbcc.ini file section, 3-12
 odbcc.ini file
 common problems, 3-9
 inspecting for problems, 3-12
 Trace parameter, 3-31

ODBCINI variable, 3-12
 OLAP Architect. *See* Hyperion Integration Server Desktop
 OLAP Builder. *See* Hyperion Integration Server Desktop
 OLAP Catalog. *See* OLAP Metadata Catalog
 OLAP Command Interface
 asynchronous commands, 5-10
 batch mode, 5-3
 command syntax, 5-6
 commands, list, 5-9
 description, 5-3
 executable name, 5-2
 executing commands, 5-6
 information, 5-1
 interactive mode, 5-4
 loading data with, 5-21
 loading members with, 5-20
 running scripts with, 5-7
 starting, 5-2
 using to test connections, 3-13
 OLAP Command Interface commands
 EXIT, 5-20
 LOADALL, 5-17
 LOADDATA, 5-15
 LOADMEMBER, 5-13
 LOGIN, 5-10
 LOGOUT, 5-19
 SETCATALOG, 5-12
 SETSOURCE, 5-11
 SETTARGET, 5-12
 SHUTDOWN, 5-19
 STATUS, 5-10
 syntax, 5-6
 VERSION, 5-11
 OLAP Command Script files, 5-8
 OLAP Integration Server
 buffer size for loads, 1-5
 common problems, 3-2
 defined, 1-1
 determining user access, 2-5
 executable name, 1-2
 log files, 1-6

- logging on to, [5-10](#), [5-20](#), [5-21](#)
- logging out of, [5-19](#)
- OLAP Metadata Catalog, information in, [4-2](#)
- port number, [1-5](#), [3-28](#)
- prerequisites for, [1-2](#)
- problems, correcting, [3-27](#)
- server problems, [3-20](#)
- shutting down, [1-6](#), [5-19](#)
- starting, [1-2](#)
- testing connection to Hyperion Essbase, [3-17](#)
- testing connection to RDBMS, [3-13](#)
- OLAP Integration Server users
 - creating, [2-2](#)
 - defined, [2-1](#)
 - deleting, [2-2](#)
 - disconnecting, [2-2](#)
 - managing, [2-2](#)
 - multiple, [2-4](#)
 - viewing, [2-2](#)
 - viewing locks, [2-4](#)
- OLAP Metadata Catalog
 - accessing, [2-1](#)
 - backups, [4-2](#)
 - connecting to, [5-21](#)
 - connection problems, [3-2](#), [3-4](#)
 - deleting locks, [2-6](#)
 - Drill-Through tables in, [4-8](#)
 - identifying before loading, [5-12](#)
 - manual editing, [4-2](#)
 - metaoutline tables, [4-5](#)
 - OLAP Integration Server and, [1-1](#)
 - OLAP model tables, [4-3](#)
 - problems, database level, [3-23](#)
 - recovery, [4-2](#)
 - users, [2-2](#)
- OLAP metaoutline. *See* metaoutline
- OLAP models
 - access permissions, [2-6](#)
 - accessing, [2-1](#)
 - locks, [2-2](#)
 - opening in exclusive access mode, [2-3](#)
 - opening in standard access mode, [2-2](#)
 - overwriting, [2-3](#)
 - permissions, viewing, [2-7](#)
 - restoring, [4-2](#)
 - tables in OLAP Metadata Catalog, [4-3](#)
 - users, [2-4](#), [2-5](#)
- olapicmd program
 - problems, [3-14](#)
 - starting, [5-2](#)
- olapisvr program
 - command-line switches, [1-3](#)
 - starting, [1-2](#)
- olapisvr.log file, specifying location, [1-4](#)
- OM_DESCRIPTION table, [4-5](#)
- OM_INFO table, [2-7](#), [4-4](#)
- OMB_DETAILS table, [4-4](#)
- OMB_INFO table, [4-4](#)
- online guide
 - accessing, [ix](#)
 - displaying, [x](#)
 - searching, [x](#)
- online help
 - accessing, [ix](#)
 - printing, [ix](#)
- opening
 - metaoutlines, options, [2-2](#)
 - OLAP models, options, [2-2](#)
- optimistic locking, [2-2](#)
- optimizing, data loads, [1-5](#)
- Oracle
 - common problems, [3-4](#)
 - tnsnames.ora file, [3-26](#)
- orphaned locks, deleting, [2-4](#), [2-6](#)
- OV_INFO table, [4-4](#)
- overwriting, OLAP models and metaoutlines,
 - problem, [2-3](#)
- OVL_REL_DETAILS table, [4-4](#)
- OVL_REL_INFO table, [4-4](#)
- OVP_REL_DETAILS table, [4-5](#)
- OVP_REL_INFO table, [4-5](#)

P

- password problems, [3-2](#), [3-21](#)
- performance, enhancing, data loads, [1-5](#)
- permissions
 - changing, [2-7](#)
 - for metaoutlines, [2-6](#)
 - for OLAP models, [2-6](#)
 - problems in the RDBMS, [3-8](#)
 - shutting down OLAP Integration Server, [5-19](#)
 - viewing, [2-6](#)
 - working with, [2-1](#)
- port 3388, [1-5](#), [3-28](#)
- prerequisites
 - for OLAP Integration Server, [1-2](#)
 - for using this guide, [vii](#)
 - OLAP Command Interface, [5-2](#)
- privileges, common problems, [3-2](#)
- problems
 - system-level, [3-20](#)
 - TCP/IP, [3-5](#)
 - with connections, correcting, [3-1](#), [3-4](#)
 - with ODBC, correcting, [3-1](#), [3-4](#), [3-24](#)
- profile file, [3-11](#)

R

- RDBMS
 - client problems, correcting, [3-24](#)
 - common problems, [3-2](#)
 - problems, correcting, [3-22](#)
 - server name problems, [3-20](#)
 - testing connection from database client, [3-7](#)
 - testing connection from Hyperion Integration Server Desktop, [3-16](#)
 - testing connection from ODBC, [3-9](#)
 - testing connection from OLAP Integration Server, [3-13](#)
 - user account problems, [3-8](#)
- read access, setting, [2-6](#)
- read locks
 - defined, [2-2](#)
 - deleting, [2-6](#)
 - viewing, [2-5](#)

- recording
 - data load scripts, [5-6](#)
- recording scripts, [5-6](#)
- recovery, OLAP Metadata Catalog, [4-2](#)
- related documentation, [viii](#)
- relational database management system. *See* RDBMS
- relational databases
 - connections to, diagram, [3-6](#)
 - problems, correcting, [3-22](#)
- releases
 - compatibility problems, [3-14](#)
 - determining for Hyperion Integration Server, [5-11](#)
- restoring
 - metaoutlines, [4-2](#)
 - OLAP Metadata Catalogs, [4-2](#)
 - OLAP models, [4-2](#)
- right-click menus, [xi](#)
- rights, problems in the RDBMS, [3-8](#)
- running
 - OLAP Command Interface, [5-2](#)
 - scripts with OLAP Command Interface, [5-7](#)

S

- scheduling, batch files for loads, [5-20](#), [5-21](#)
- scripts
 - incremental loads, syntax, [5-9](#)
 - member and data loads, recording, [5-6](#)
 - running with OLAP Command Interface, [5-7](#)
- security
 - access permissions, [2-6](#)
 - locks, [2-2](#)
 - problems, [3-5](#)
- semicolons, in OLAP Command Interface syntax, [5-6](#)
- server problems, [3-20](#)
- server software
 - port number, [3-28](#)
 - problems, correcting, [3-27](#)
 - troubleshooting, [3-1](#)

Servers menu, OLAP Metadata Catalog, Delete

Locks command, [2-4](#)

SET, [5-4](#)

SET commands, [5-4](#)

SETCATALOG command

described, [5-12](#)

example of, [5-21](#)

SETSOURCE command

described, [5-11](#)

example of, [5-20](#), [5-21](#)

SETTARGET command

described, [5-12](#)

example of, [5-20](#), [5-21](#)

setting

data source, [5-11](#), [5-20](#), [5-21](#)

Essbase server, [5-20](#)

Hyperion Essbase server, [5-21](#)

network listeners, [1-5](#)

OLAP Metadata Catalog, [5-12](#), [5-21](#)

target Hyperion Essbase data server, [5-12](#)

SHUTDOWN command, [1-6](#), [5-19](#)

shutting down, OLAP Integration Server, [1-6](#), [5-19](#)

software components

common problem areas, [3-5](#)

connection problems, [3-2](#)

SQL Server, common problems, [3-4](#)

standard access mode, [2-2](#)

starting

OLAP Command Interface, [5-2](#)

OLAP Integration Server, methods, [1-2](#)

OLAP Integration Server, programs necessary, [1-2](#)

STATUS command, described, [5-5](#), [5-10](#)

stopping, OLAP Integration Server, [1-6](#)

storage

metaoutline, [4-2](#)

OLAP Integration Server users, [2-2](#)

OLAP model, [4-2](#)

switches, for olapisvr, [1-3](#)

syntax, OLAP Command Interface commands, [5-6](#)

system variables. *See* environment variables

system-level problems, [3-20](#)

T

tables

Drill-Through, [4-8](#)

metaoutline, [4-5](#)

miscellaneous, [4-9](#)

OLAP model, [4-3](#)

read-only, [4-2](#)

TCP/IP, problems, [3-5](#)

testing

connections, overview, [3-6](#)

database client, [3-7](#)

Hyperion Integration Server Desktop connection to RDBMS, [3-16](#)

Hyperion Integration Server Desktop to Hyperion Essbase, [3-19](#)

ODBC connection to RDBMS, [3-9](#)

ODBC on UNIX, [3-11](#)

OLAP Integration Server connection to RDBMS, [3-13](#)

OLAP Integration Server connection to Hyperion Essbase, [3-17](#)

timestamp, locks and, [2-4](#)

tnsnames.ora file, [3-26](#)

tnsping program, [3-9](#)

Trace parameter, odbci.ini file, [3-31](#)

tracing utility

ODBC for UNIX, [3-31](#)

ODBC for Windows, [3-30](#)

training services, [xi](#)

troubleshooting

connections, [3-1](#)

ODBC, [3-1](#)

system-level problems, [3-20](#)

turning off, OLAP Integration Server, [1-6](#)

typographical conventions, [x](#)

U

UNIX

ODBC tracing, [3-31](#)

testing ODBC, [3-11](#)

unreleased locks, deleting, [2-4](#), [2-6](#)

user accounts, administrative, [3-8](#)

- user names
 - locks and, [2-4](#)
 - problems with, [3-2](#), [3-21](#)
- users
 - deleting, [2-2](#)
 - disconnecting, [2-2](#)
 - Hyperion Essbase, defined, [2-1](#)
 - managing permissions and locks, [2-1](#)
 - multiple, [2-4](#)
 - OLAP Integration Server, defined, [2-1](#)
 - OLAP Integration Server, managing, [2-2](#)

V

- variables. *See* environment variables
- VERSION command, described, [5-5](#), [5-11](#)
- versions, compatibility problems, [3-14](#)
- viewing
 - log files, [1-6](#)
 - permissions for OLAP models, [2-7](#)

W

- warning messages, [1-4](#)
- Windows
 - ODBC tracing, [3-30](#)
 - testing ODBC, [3-10](#)
- write access, setting, [2-6](#)
- write locks
 - defined, [2-2](#)
 - deleting, [2-6](#)
 - viewing, [2-5](#)

