Integration Services System Administrator’s Guide, 9.3.1

Copyright © 1998, 2007, Oracle and/or its affiliates. All rights reserved.

Authors: David Lang

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are “commercial computer software” or “commercial technical data” pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee’s responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.
Contents

Chapter 1. Using Essbase Integration Server ................................................ 7

Integration Server .......................................................................................... 7
Starting Integration Server .............................................................................. 7
Integration Server Startup Switches ............................................................... 10
Storing Startup Information in the Configuration File ................................. 20
Storing Startup Information in the Startup File (Windows Only) ................ 21
Additional Configuration Parameters .......................................................... 22
Reconnecting .................................................................................................. 23
Shutting Down Integration Server .................................................................. 24
Integration Server Log File ........................................................................... 25
Items Logged in Integration Server File ........................................................ 25
Integration Server Log File Size .................................................................... 26
Data Load Error File ....................................................................................... 26
Unicode .......................................................................................................... 27
Unicode and Non-Unicode Application Modes .............................................. 27
When to Use Unicode-Mode Applications ...................................................... 28
Unicode-Enabled Administration Tools ......................................................... 29
Analytic Integration Connector for SAP BW .................................................. 29

Chapter 2. Working with Users, Locks, and Permissions ............................... 31

Working with Users ....................................................................................... 31
Working with Locks ....................................................................................... 32
Using Standard Access Mode ....................................................................... 32
Using Exclusive Access Mode ..................................................................... 32
Correcting Problems with Locks ................................................................... 33
Viewing Integration Services Users with Locks .......................................... 33
Deleting Locks for Integration Services Users .............................................. 34
Working with Permissions ............................................................................. 35
Viewing Permissions for OLAP Models ......................................................... 35
Viewing Permissions for Metaoutlines .......................................................... 36
### Chapter 3. Troubleshooting ODBC and Connections

- Common Connection Problems ........................................... 39
- Database-Specific Problems ............................................. 41
- ODBC and Connection Problems ........................................ 41
- Isolating Connection Problems ......................................... 42
  - Testing the Database Client Connection to the Data Source .......... 44
  - Testing the ODBC Connection to the Data Source ...................... 45
  - Testing the Integration Server Connection to the Data Source .......... 49
  - Testing the Integration Services Console Connection to the Data Source .... 51
  - Testing the Integration Server Connection to Essbase Server .......... 52
  - Testing the Integration Services Console Connection to Essbase Server .......... 53
- Correcting Connection Problems ......................................... 54
  - User Name and Password Problems ..................................... 55
  - Correcting Data Source Problems ....................................... 56
  - Correcting Database Client Problems ................................... 56
  - Correcting ODBC Problems .............................................. 58
  - Correcting Integration Server Problems ................................ 59
  - Correcting Integration Services Console Problems ...................... 60
- Using ODBC Tracing ..................................................... 61
  - Using ODBC Tracing on Windows Systems .............................. 61
  - Using ODBC Tracing on UNIX Systems .................................. 61

### Chapter 4. Tables in OLAP Metadata Catalog

- OLAP Metadata Catalog .................................................. 63
- Tables Relating to the OLAP Model ..................................... 64
- Tables Relating to the Metaoutline ...................................... 65
- Tables Relating to Drill-Through ........................................ 67
- Tables Relating to Hybrid Analysis ..................................... 67
- Miscellaneous Tables ..................................................... 68

### Chapter 5. Using Integration Services Shell

- Starting Integration Services Shell ..................................... 69
- Integration Services Shell ................................................. 70
  - SET and LOAD Commands .............................................. 71
  - Informational Commands ................................................ 72
  - Command Syntax and Execution ....................................... 72
  - Recording Member and Data Load Scripts .............................. 73
  - Running Integration Services Shell Script and Batch Files .......... 73
  - Running OLAP Command Script Files .................................. 74
- Integration Services Shell Commands .................................. 75
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOGIN</td>
<td>75</td>
</tr>
<tr>
<td>STATUS</td>
<td>76</td>
</tr>
<tr>
<td>VERSION</td>
<td>76</td>
</tr>
<tr>
<td>SETLOCALE</td>
<td>76</td>
</tr>
<tr>
<td>SETSOURCE</td>
<td>77</td>
</tr>
<tr>
<td>SETTARGET</td>
<td>78</td>
</tr>
<tr>
<td>SETCATALOG</td>
<td>78</td>
</tr>
<tr>
<td>LOADMEMBER</td>
<td>78</td>
</tr>
<tr>
<td>LOADDATA</td>
<td>81</td>
</tr>
<tr>
<td>LOADALL</td>
<td>83</td>
</tr>
<tr>
<td>LOGOUT</td>
<td>87</td>
</tr>
<tr>
<td>SHUTDOWN</td>
<td>87</td>
</tr>
<tr>
<td>EXIT</td>
<td>88</td>
</tr>
<tr>
<td>Performing Member Loads Using Integration Services Shell</td>
<td>88</td>
</tr>
<tr>
<td>Loading Data Using Integration Services Shell</td>
<td>88</td>
</tr>
<tr>
<td>Chapter 6. Naming Restrictions for Essbase Applications, Databases, and Members</td>
<td>91</td>
</tr>
<tr>
<td>Naming Restrictions for Applications and Databases</td>
<td>91</td>
</tr>
<tr>
<td>Naming Restrictions for Dimensions, Members, and Aliases</td>
<td>92</td>
</tr>
<tr>
<td>Using Dimension and Member Names in Calculation Scripts, Report Scripts, Formulas, and Filters</td>
<td>94</td>
</tr>
<tr>
<td>Appendix A. Return Codes and Error Messages</td>
<td>95</td>
</tr>
<tr>
<td>OLAPICMD Session Return Codes</td>
<td>95</td>
</tr>
<tr>
<td>Member Load Error Messages</td>
<td>96</td>
</tr>
<tr>
<td>Data Load Error Messages</td>
<td>99</td>
</tr>
<tr>
<td>Drill-Through Report Error Messages</td>
<td>100</td>
</tr>
<tr>
<td>Miscellaneous Error Messages</td>
<td>101</td>
</tr>
<tr>
<td>Essbase Error Messages Generated During Data Loads</td>
<td>103</td>
</tr>
<tr>
<td>Appendix B. Integration Services Limits and Guidelines</td>
<td>105</td>
</tr>
<tr>
<td>Artifact Limits</td>
<td>105</td>
</tr>
<tr>
<td>Source Database Artifact Naming Limits</td>
<td>108</td>
</tr>
<tr>
<td>Setting the Compression Dimension</td>
<td>109</td>
</tr>
<tr>
<td>Guidelines</td>
<td>109</td>
</tr>
<tr>
<td>Hybrid Analysis Guidelines</td>
<td>110</td>
</tr>
<tr>
<td>Data Source Guidelines</td>
<td>110</td>
</tr>
<tr>
<td>Dimensions Guidelines</td>
<td>110</td>
</tr>
<tr>
<td>Members Guidelines</td>
<td>110</td>
</tr>
<tr>
<td>Spreadsheet Guidelines</td>
<td>111</td>
</tr>
</tbody>
</table>
This chapter describes starting and stopping Essbase Integration Server and how to view its log file.

Integration Server

Integration Server is multithreaded server software that is the bridge between the data source, OLAP Metadata Catalog, and Essbase Server. Oracle's Essbase® Integration Services performs several tasks:

- Retrieves OLAP model and metaoutline information from OLAP Metadata Catalog
- Generates SQL statements
- Retrieves data from external sources
- Loads members and data into the Oracle's Hyperion® Essbase® – System 9 database

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

Starting Integration Server

Before starting Integration Server, verify all required components are running. These components need not be on your local computer.

- If you are accessing relational data sources, verify these components are running:
The relational database management system (RDBMS) with the relational data source containing the OLAP Metadata Catalog where the metadata is stored

One or more RDBMSs containing the relational data sources you want to use to create or modify an OLAP model or metacomment

- If you are accessing an SAP Business Information Warehouse (BW) data warehouse, verify Hyperion® System™ 9 BI+™ Analytic Integration Connector for SAP BW™ is running.

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

You can start Integration Server using several methods:

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

**Note:**

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

**To start Integration Server from DOS:**

1. **At the command line, type the startup.bat command.**

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

2. **If you want to change a default startup switch, add the switch to the ais.cfg file or the startup.bat file.**

3. **If you want to change a default configuration parameter, add the parameter to the ais.cfg file.**

   See “Storing Startup Information in the Configuration File” on page 20 and “Storing Startup Information in the Startup File (Windows Only)” on page 21 for information on adding startup switches and configuration parameters to files.

**To start Integration Server from UNIX:**

1. **Type is.sh.**

2. **Type the executable name on the command line, for example, type olapisvr**

   **Note:**

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

---

8 Using Essbase Integration Server
If you want to change a default startup switch, type the command syntax after the \texttt{olapisvr} command or add the switches to the \texttt{ais.cfg} file.

See “Storing Startup Information in the Configuration File” on page 20 and “Storing Startup Information in the Startup File (Windows Only)” on page 21 for information on adding startup switches to these files.

Values for startup switches entered at the command line override startup switch values stored in the \texttt{ais.cfg} file.

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

\begin{verbatim}
olapisvr -E"c:\my logs\aislog" -M"c:\ess\bin\essbase.mdb"
\end{verbatim}

To start Integration Server from the Windows Desktop Start menu:

1. Click Start.
2. Select Programs > Hyperion > Integration Services > Integration Server.

If startup switch settings are stored in \texttt{startup.bat} or \texttt{ais.cfg}, the settings in \texttt{startup.bat} override the settings in \texttt{ais.cfg} when Integration Server is started from the Windows Desktop Start menu.

To start Integration Server as a Windows service:

1. Access the Services dialog box.
   - On Windows XP, select Start > Control Panel. In the Control Panel, select Performance & Maintenance, and then Administrative Tools. In the Administrative Tools window, double-click Services.

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

2. Select, but do not double-click, Essbase Integration Server.
3. If you want to change any default startup switches or configuration parameters, add the switches to the \texttt{ais.cfg} file or the \texttt{startup.bat} file.
4. Start the service for Integration Server.
   - On Windows XP, click Start.

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

Table 1 lists the switches and configuration parameters used to change default settings when starting Integration Server from the command line, to add startup switch information to the startup.bat file, or to store startup switch information in the ais.cfg file.

### Table 1  Essbase Integration Server Switches

<table>
<thead>
<tr>
<th>switch</th>
<th>meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>-?</td>
<td>To view a list of available switches for starting Integration Server.</td>
</tr>
<tr>
<td>-A</td>
<td>To ignore the parent of a given member when updating OLAP intersections during a drill-through operation.</td>
</tr>
<tr>
<td>-C</td>
<td>To ignore parents when updating OLAP intersections, do one of the following:</td>
</tr>
<tr>
<td>-D</td>
<td>By default, Integration Server sets the value at N so that parents are not ignored when updating OLAP intersections. If a parent is ignored during an update of OLAP intersections, subsequent drill-through operations cannot be performed on any shared members related to that parent.</td>
</tr>
<tr>
<td>-E</td>
<td>In Windows:</td>
</tr>
<tr>
<td></td>
<td>In UNIX:</td>
</tr>
<tr>
<td>-F</td>
<td>In Windows, add -AY to the startup.bat file.</td>
</tr>
<tr>
<td>-I</td>
<td>In UNIX, type -AY when you start Integration Server.</td>
</tr>
<tr>
<td>-L</td>
<td>In Windows or UNIX, add [A]=Y to the ais.cfg file.</td>
</tr>
<tr>
<td>-M</td>
<td></td>
</tr>
<tr>
<td>-N</td>
<td></td>
</tr>
<tr>
<td>-P</td>
<td></td>
</tr>
<tr>
<td>-R</td>
<td></td>
</tr>
<tr>
<td>-S</td>
<td></td>
</tr>
<tr>
<td>[V]</td>
<td></td>
</tr>
<tr>
<td>-T</td>
<td></td>
</tr>
</tbody>
</table>

For information on storing startup switches in startup.bat or ais.cfg, see “Storing Startup Information in the Configuration File” on page 20 and “Storing Startup Information in the Startup File (Windows Only)” on page 21.

The following topics describe the default switches.

-? 
To view a list of available switches for starting Integration Server.

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

To ignore parents when updating OLAP intersections, do one of the following:

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

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

In Windows:

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

"C:\Hyperion\AIS\bin\olapisvr.exe" -AY

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

```
olapisvr -AY
```

**In Windows or UNIX:**

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

```
[A]=Y
```

**-B**

**Note:**

- B is no longer available.

**-C**

To set the number of records that Integration Server commits to Essbase:

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

The default setting is all records.

- C can be used with the -N switch.

**Note:**

- C should not be used in operations involving aggregate storage.

- C is often used in testing environments. The recommended setting for testing purposes is 5,000 to 10,000 records. After you have completed testing, shut down Integration Server to revert to the default setting, or reset to the limit allowed by DATAERRORLIMIT.

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

**Note:**

If you have included a setting for testing purposes for -C in ais.cfg, be sure to edit the file to clear the `ais.cfg` file so that it will change the setting back to “all records.”
Setting a smaller number of records during testing enables you to see data load errors incrementally, providing you an opportunity to correct problems before continuing. Resetting the number of records to the default of all records results in faster performance.

There is no limit to the number of records that Integration Server can transfer to Essbase during a data load. If, however, you experience memory problems during a data load, set -C to a smaller value.

**Note:**

The setting of -C does not apply to member loads.

**In Windows:**

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

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

**In UNIX:**

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

```
olapisvr -C5000
```

**In Windows or UNIX:**

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

```
[C]=5000
```

**-D**

To specify whether or not to add the DISTINCT clause to the SELECT statement of drill-through SQL.

The default behavior adds the DISTINCT clause. To not add the DISTINCT clause:

- In Windows, add `-DN` to the `startup.bat` file.
- In UNIX, type `-DN` when you start Integration Server.
- In Windows or UNIX, add `[D]=N` to the `ais.cfg` file.

**In Windows:**

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

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

Any value other than -DN or no value specifies that the DISTINCT clause should always be added to drill-through SQL.
In UNIX:
To specify that the DISTINCT clause should not be added to drill-through queries, type:
olapisvr -DN

In Windows or UNIX:
In the ais.cfg file, this parameter is specified in the following format:
[D]=N

-E
To give the log file a different name:
- In Windows, add -Elog_file_name to the startup.bat file.
- In UNIX, type -Elog_file_name when you start Integration Server.
- In Windows or UNIX, add [E]=log_file_name to the ais.cfg file.
The .log extension is added automatically. The default name and location is ISHOME\log\olapisvr.log.

In Windows:
In the startup.bat file, to name the log file myserver.log and put it in the temp directory on Windows, add -Ec:\temp\myserver after "C:\Hyperion\AIS\bin\olapisvr.exe". For example:
"C:\Hyperion\AIS\bin\olapisvr.exe -E"c:\temp\myserver
Do not type the .log file extension. The .log extension is automatically appended to the name you type for the log.
Always enclose path names that contain spaces with quotation marks (" "); for example, type:
"C:\Hyperion\AIS\bin\olapisvr.exe" -I -E"c:\program files\ais\myserver"

In UNIX:
To name the log file myserver.log and put it in the temp directory on Windows, type
olapisvr -Ec:\temp\myserver
Enclose path names that contain spaces in quotation marks (" "); for example, type:
olapisvr -E"c:\program files\ais\myserver"

In Windows or UNIX:
In the ais.cfg file, this parameter is specified in the following format:
[E]=c:\temp\myserver
or
[E]=c:\program files\ais\myserver
Note:
In the ais.cfg file, you do not need to type quotation marks around path names that contain spaces.

-F
To specify whether to add parentheses around user-defined drill-through filters. See Essbase Spreadsheet Add-in help.
The default is not to add parentheses around user-defined drill-through filters. To add parentheses:
- In Windows, add -FY to the startup.bat file.
- In UNIX, type -FY when you start Integration Server.
- In Windows or UNIX, add [F]=Y to the ais.cfg file.

In Windows:
In the startup.bat file, to specify that parentheses should be added around user-defined drill-through filters, add -FY after "C:\Hyperion\AIS\bin\olapisvr.exe". For example:
"C:\Hyperion\AIS\bin\olapisvr.exe" -FY
Any other value or no value specifies that parentheses should not be added around user-defined drill-through filters.

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

In Windows or UNIX:
In the ais.cfg file, this parameter is specified in the following format:
[F]=Y

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

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

Note:
-I cannot be entered in the ais.cfg file. It can only be specified in the startup.bat file.
In Windows:
In the startup.bat file, to prevent Integration Server from shutting down as a service, add -I after "C:\Hyperion\AIS\bin\olapisvr.exe". For example:
"C:\Hyperion\AIS\bin\olapisvr.exe" -I

-L
To set the level of detail of the messages that Integration Server logs:
● In Windows, add -Llevel to the startup.bat file.
● In UNIX, type -Llevel when you start Integration Server.
● In Windows or UNIX, add -Llevel to the ais.cfg file.
The default level is 2. You can specify the following levels:

0 To log all debug messages and the ODBC SQL generated by Integration Server; generate the dataloadrecord.txt file, listing the first 1,000 records, which may include successfully loaded records and rejected records; and create the hisdld.rul file consisting of the names of the dimensions loaded followed by the leaf node number of each dimension or member

2 To log all informational messages and some of the ODBC SQL generated by Integration Server

3 To log all warnings

4 To log all noncritical errors

7 To log all critical errors where Integration Server terminates the command

8 To log all critical server errors

The recommended level setting is 2 or 3.

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

In UNIX:
To use the highest level of logging, type:
olapisvr -L0

In Windows or UNIX:
In the ais.cfg file, this switch is specified in the following format:
[L]=0

-M
To specify the location of the Essbase message database file if the file is moved to a new location:
In Windows, add `-M path\essbase.mdb` to the `startup.bat` file.

In UNIX, type `-M path\essbase.mdb` when you start Integration Server.

In Windows, add `-M path\essbase.mdb` to the `ais.cfg` file.

The default location is `ISHOME\bin\essbase.mdb`.

In Windows:

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

```
"C:\Hyperion\AIS\bin\olapisvr.exe" -M c:\essbasemsg\bin\essbase.mdb
```

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

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

In UNIX:

To specify the new location of `essbase.mdb` as `essbasemsg\bin`, type:

```
olapisvr -M c:\essbasemsg\bin\essbase.mdb
```

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

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

In Windows or UNIX:

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

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

or

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

**Note:**

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

**-N**

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

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

The default setting is 1 thread.

-`N` may be used with `-C`.  

---

16 Using Essbase Integration Server
—N controls the number of threads allocated to data load optimization. Usually, users start with 2 threads and adjust according to the environment.

**In Windows:**

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

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

**In UNIX:**

To set the number of threads to 2, type:

```
olapisvr -N2
```

**In Windows or UNIX:**

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

```
[N]=2
```

**-P**

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

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

**In Windows:**

In the `startup.bat` file, add `-Pportnumber` after "C:\Hyperion\AIS\bin\olapisvr.exe". For example:

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

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

```
aspen:8850
```

When starting Integration Services Shell, to change the TCP port number, at the command line, type `-Pportnumber`. For example:

```
olapicmd -P8850
```

**In UNIX:**

When starting Integration Server or Integration Services Shell, to change the TCP port number, type:

```
olapisvr -P8850
```

or

```
olapicmd -P8850
```
In Windows or UNIX:

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

\[P\]=8850

-\textbf{Q}

\textbf{Note:}

-\textit{Q} is no longer available.

-\textbf{R}

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

-\textbullet\ When -R is not set, users have unrestricted access to all table records.
-\textbullet\ When -R is set to Y, users have no access to any table records.
-\textbullet\ When -R is set to F, users have no access to fact table records but have unrestricted access to all other table records.

To restrict users from browsing fact table records:

-\textbullet\ In Windows, add -R\textit{access\_parameter} to the startup.bat file.
-\textbullet\ In UNIX, type -R\textit{access\_parameter} when you start Integration Server.
-\textbullet\ In Windows or UNIX, add [R]=\textit{access\_parameter} to the ais.cfg file.

By default, -R is not set, giving users unrestricted access to all table records.

In Windows:

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

"C:\Hyperion\AIS\bin\olapisvr.exe" -RY

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

"C:\Hyperion\AIS\bin\olapisvr.exe" -RF

In UNIX:

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

olapisvr -RY

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

olapisvr -RF

In Windows or UNIX:

In the ais.cfg file, this parameter is specified in the following format:
To specify the stack size in megabytes of each thread that Integration Server creates:

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

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

**In Windows:**

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

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

**In UNIX:**

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

```
olapisvr -S1
```

**In Windows or UNIX:**

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

```
[S]=1
```

To set the number of network listeners that Integration Server starts:

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

The default (and recommended) number is 10.

Network listeners receive requests from Integration Services Console. Integration Server automatically adds and subtracts listeners as needed, so the number of listeners set at startup does not limit the number of users that can connect to an Integration Server.

**In Windows:**

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

```
"C:\Hyperion\AIS\bin\olapisvr.exe" -T17
```
In UNIX:
To set the number of network listeners Integration Server starts to 17, type:
olapisvr -T17

In Windows or UNIX:
In the ais.cfg file, -T is specified in the following format:
[T]=17

-U
To specify whether the Essbase database and application should be unloaded from memory after a load is completed.
Integration Server can successfully unload the Essbase database from memory only if there are no users accessing it.
To specify whether the Essbase database and application should be unloaded from memory after a load is completed, do one of the following:
● In Windows, add -UY to the startup.bat file.
● In UNIX, type -UY when you start Integration Server.
● In Windows or UNIX, add [U]=Y to the ais.cfg file.
By default, Integration Server does not unload the Essbase database from memory after a load is completed.

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

In UNIX:
To specify whether the Essbase database and application should be unloaded from memory after a load is completed, type
olapisvr -UY

In Windows or UNIX:
In the ais.cfg file, -U is specified in the following format:
[U]=Y

Storing Startup Information in the Configuration File
The ais.cfg file has two purposes:
● To configure the required JDBC drivers for Teradata and Oracle users. See the Essbase Integration Services Installation Guide

20 Using Essbase Integration Server
To pass startup switches and configuration parameters, other than the defaults, when launching Integration Server

The ais.cfg file is a text file residing in the \bin directory on Windows, UNIX, and Linux platforms. When you start Integration Server from the command line using the startup.bat command (in DOS or UNIX), the startup routine checks the contents of ais.cfg for any startup switch or parameter information. Then one of the following applies:

- Startup switch or configuration parameter syntax added to the ais.cfg file override default settings (see “Integration Server Startup Switches” on page 10 and “Storing Startup Information in the Configuration File” on page 20).
- If you did not add startup switch or parameter information to ais.cfg, the default settings are used to start Integration Server.
- If ais.cfg contains startup switch syntax and you enter startup switch overrides in the startup.bat file (Windows only) any startup switches in startup.bat override both the startup switch settings contained in the ais.cfg file and the default settings.

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

1 Open the ais.cfg file in any text editor.

2 Add the startup switch you want to store by typing the startup switch letter in brackets; for example:
   ```
   [B]
   ```

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

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

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

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

4 Save and close ais.cfg.

---

**Storing Startup Information in the Startup File (Windows Only)**

The startup.bat file has two purposes:

- To configure the common Hyperion components, such as Java, ODBC, and JDBC drivers, for Teradata and Oracle users. See the Essbase Integration Services Installation Guide
- To pass startup switches and configuration parameters, other than the defaults, when launching Integration Server

The startup.bat file is a text file residing in the \bin directory. When you start Integration Server as a Windows service, the startup routine checks the contents of startup.bat for any startup switch information. Then the startup routine checks ais.cfg for any startup switch and configuration parameter information. Then one of the following applies:
● Startup switch syntax added to the `startup.bat` file (Windows only), override default settings (see “Integration Server Startup Switches” on page 10).

● If you added startup switch or configuration parameter syntax only to the `ais.cfg` file, those settings that you added override the default settings for those switches and parameters (see “Integration Server Startup Switches” on page 10 and “Storing Startup Information in the Configuration File” on page 20).

● If you did not add startup switch or parameter information to `startup.bat` or `ais.cfg`, the default settings (see “Integration Server Startup Switches” on page 10 and “Storing Startup Information in the Configuration File” on page 20) are used to start Integration Server.

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

1 Open the `startup.bat` file in any text editor.

2 If you want to change a default startup switch, type the command syntax in the line that calls the Integration Server.

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

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

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

   **Note:**

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

   You can only enter startup switch information into `startup.bat`. Configuration parameters must be entered in `ais.cfg`.

3 Save and close `startup.bat`.

### Additional Configuration Parameters

Add the configuration parameters specified in this topic to the `ais.cfg` file to change the default configuration settings described in “Storing Startup Information in the Configuration File” on page 20.

### Windows Configuration Parameters

The configuration parameters described in this section can only be specified in the `ais.cfg` file that is installed on Windows. These parameters cannot be specified in the `startup.bat` file, at the command line, or in the `ais.cfg` file that is installed with the UNIX version of Integration Server.
[K]
To specify whether to display the primary keys in an OLAP model. The default behavior is not to display the primary keys in an OLAP model.

If you want to display the primary keys in an OLAP model, add \([K]=Y\) to the \(\text{ais.cfg}\) file.

**In Windows:**
In the \(\text{ais.cfg}\) file, this parameter is specified in the following format:
\([K]=Y\)

[V]
To specify whether Integration Server automatically validates an OLAP model or metaoutline when a Save or Save As operation is performed. The default behavior is for Integration Server to validate an OLAP model or metaoutline before saving it.

If you want to disable automatic OLAP model and metaoutline validation, add \([V]=0\) to the \(\text{ais.cfg}\) file.

**In Windows:**
In the \(\text{ais.cfg}\) file, this parameter is specified in the following format:
\([V]=0\)

**Windows and UNIX Configuration Parameter for Aggregate Storage**
The configuration parameter described in this section can be specified in the \(\text{ais.cfg}\) file that is installed on Windows and UNIX. This parameter cannot be specified in the \(\text{startup.bat}\) file at the command line.

[ADDDUPLICATESFORASO]
To allow addition of duplicate data records when using an aggregate storage database. The default behavior is to set the data load option in aggregate storage to not allow duplicate records.

If you want to allow duplicate data records when using an aggregate storage database, add \([ADDDUPLICATESFORASO]=Y\) to the \(\text{ais.cfg}\) file.

**In Windows and UNIX:**
In the \(\text{ais.cfg}\) file, this parameter is specified in the following format:
\([ADDDUPLICATESFORASO]=Y\)

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

2. In the Integration Services Console, select Connections > OLAP Metadata Catalog > Reconnect.

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

Shutting Down Integration Server

After using Integration Server, shut it down using any of several methods.

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

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

- On Windows platforms, close the Integration Server window.
- Issue the SHUTDOWN command in the Integration Services Shell.
See “Integration Services Shell Commands” on page 75.

To shut down Integration Server from the Windows desktop:

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

To shut down Integration Server as a Windows service:

1. Select Start > Settings > Control Panel.
2. Open the Services dialog box.
   - On Windows 2000 and 2003, double-click Administrative Tools, and then double-click Services.
   - On Windows XP, select Performance & Maintenance, and then Administrative Tools. In the Administrative Tools window, double-click Services.
3. Locate Essbase Integration Server in the list of services.
4. Stop the service.
   - On Windows XP, right-click Essbase Integration Server and select Stop.
Integration Server Log File

By default, the Integration Server log file is named \olapisvr.log and is located in the ISHOME\log directory. View it from Integration Services Console by selecting Tools > View Log File to launch the Server Log file dialog box, or open \olapisvr.log in any text editor. To create the log file with a different location or name, use -E. See “Integration Server Startup Switches” on page 10.

Items Logged in Integration Server File

Items are logged in the Integration Server log file, depending on the logging level you set when you start Integration Server:

- Processing messages
- Member load error messages
- Settings for switches used to start Integration Server from the command line or as a Windows service
- Two copies of each SQL statement generated to access the data source as follows:
  - The first statement is in the SQL dialect understood by the ODBC driver and is generated by Integration Server.
  - The second statement is generated by the ODBC driver and is translated by the driver into the dialect of SQL understood by the data source being accessed.
- Integration Services Shell commands used to perform an action
- Connection information
- Load status
- Names of any metaoutline running
- Other informational messages

When a user accesses a drill-through report using Essbase Spreadsheet Add-in, items related to the drill-through report are also logged:

- Drill-through report name
- Metaoutline with which this drill-through report is associated
- Essbase Server computer being accessed
- Essbase application and database
- User accessing drill-through report

See “Integration Server Startup Switches” on page 10.

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

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

Your server log file will grow to a maximum of 50 megabytes in size. Then it will be truncated to 10 megabytes with the most recent 10 megabytes of entries being retained. The log file will again grow to 50 megabytes before it is truncated.

Change the default settings by specifying LOGFILEMIN and LOGFILEMAX in the ais.cfg file, for example:

```plaintext
[LOGFILEMIN] = 20
[LOGFILEMAX] = 60
```

Data Load Error File

If data loads generate errors, the file `dataload.txt` is placed in a folder Integration Server creates for the load under the `ISHOME/loadinfo` directory. The `dataload.txt` file lists rejected dimensions and members and error message codes to identify data load problems.

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

```plaintext
<application_database_timestamp_sessionnumber>
```

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

```plaintext
MyTBC_MyTBC_DB_2003_May_1_10_00_pm_<sessionnumber>
```

In the case of data load failure, open the `dataload.txt` file located in the folder described previously and review the error codes.

These are the most common error codes:

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

Figure 1 shows three records from the `dataload.txt` file with the error code 3303.
Sharing data across national and language boundaries is a challenge for multi-national businesses. Traditionally, each computer stores and renders text based on its locale specification. A locale identifies the local language and cultural conventions such as the formatting of currency and dates, sort order of the data, and the character set encoding to be used on the computer. The encoding of a character set refers to the specific set of bit combinations used to store the character text as data, as defined by a code page or an encoding format. In Essbase, code pages map characters to bit combinations for non-Unicode encodings.

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

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

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

User-defined character sets (UDC) are not supported and the Chinese National Standard GB 18030-2000 is not supported.

Note:
For information on using Unicode in Essbase, see the Essbase Services Database Administrator’s Guide.

Unicode and Non-Unicode Application Modes
Applications are designated as Unicode-mode applications or non-Unicode-mode applications. Unicode-mode applications support multiple character sets. Integration Services uses the UTF-8 encoding form to interpret and store character text. Character-based artifacts in Unicode-mode applications, such as member and alias names, can include characters from different languages. Clients working with Unicode-mode applications can have different locales than Essbase Server. For example, client computers with Japanese locales and client computers with German locales...
can work with the same Unicode-mode application on an Essbase Server that has a Spanish locale.

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

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

**When to Use Unicode-Mode Applications**
Consider working with Unicode-mode applications only if you have any of the following situations:

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

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

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

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

Analytic Integration Connector for SAP BW

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

Integration Services functions with Analytic Integration Connector for SAP BW through BAPI Protocol for online analytic processing (OLAP). The open database connectivity (ODBC) driver used for this process is manufactured by Simba Technologies. (See the Hyperion Analytic Integration Connector Installation Guide.)
Working with Users, Locks, and Permissions

In This Chapter

- Working with Users ......................................................... 31
- Working with Locks ............................................................... 32
- Working with Permissions ...................................................... 35

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.

See Chapter 4, “Tables in OLAP Metadata Catalog.”

Working with Users

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

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

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

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

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

Tip:

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

Using Standard Access Mode

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

When you save an OLAP model or metaoutline in standard access mode, Integration Services upgrades the read lock to a write lock for the duration of the save. After Integration Services saves the changes to the OLAP model or metaoutline, the lock reverts back to a read lock. However, if other users have the OLAP model or metaoutline open, your changes are not updated for the other users until they reopen the document.

In addition, if other users have the OLAP model or metaoutline open at the same time as you do, they can edit and save it, thereby overwriting your changes. Using exclusive access mode prevents users from overwriting your changes. See “Using Exclusive Access Mode” on page 32.

➤ If you 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 actions:
  ● Save the OLAP model or metaoutline using a different name. See Integration Services Console Help.
  ● If you have not yet started editing the document or have entered only a few changes, close the OLAP model or metaoutline and then reopen it using exclusive access mode and reapply the changes.
  ● Contact the users with read locks and ask them to close the OLAP model or metaoutline. To view a list of users with read locks, see “Viewing Integration Services Users with Locks” on page 33.

Using Exclusive Access Mode

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

**Correcting Problems with Locks**

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

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

If you suspect that unreleased locks remain from other user sessions, check to see which users have locks. See “Viewing Integration Services Users with Locks” on page 33. If you are convinced that some or all of the locks are from terminated user sessions, delete them. See “Deleting Locks for Integration Services Users” on page 34.

**Viewing Integration Services Users with Locks**

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

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

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

   ```sql
   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.

<table>
<thead>
<tr>
<th>CB_USER_NAME</th>
<th>CB_OBJECT_ID</th>
<th>CB_OBJECT_TYPE</th>
<th>CB_OBJECT_LOCK</th>
<th>CB_LOCK_TIMESTAMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>sa</td>
<td>889844639</td>
<td>1</td>
<td>1</td>
<td>Apr 9 2004 4:43PM</td>
</tr>
<tr>
<td>sa</td>
<td>889845263</td>
<td>2</td>
<td>1</td>
<td>Apr 9 2004 4:43PM</td>
</tr>
<tr>
<td>sa</td>
<td>889844639</td>
<td>1</td>
<td>1</td>
<td>Apr 9 2004 5:20PM</td>
</tr>
<tr>
<td>sa</td>
<td>892167813</td>
<td>2</td>
<td>1</td>
<td>Apr 9 2004 5:20PM</td>
</tr>
</tbody>
</table>

   (4 row(s) affected)

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

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

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

### Deleting Locks for Integration Services Users

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

**Caution!**

Make sure the users do not need their locks before you delete the locks.
To delete read and write locks on OLAP models or metaoutlines:

1. **Determine which users have locks.**
   
   See “Viewing Integration Services Users with Locks” on page 33.

2. **Delete the rows containing the unwanted locks.**

   For example, to delete all locks held by the `sa` user, issue the following DELETE statement in the OLAP Metadata Catalog database, using the tools for the data source:
   
   ```sql
   DELETE FROM CB_CONCURRENCY WHERE CB_USER_NAME = 'sa'
   ```

### Working with Permissions

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

- **Allow read/write access for other users.** This setting permits all other users to read and write to the OLAP model or metaoutline. This setting is the default.

- **Allow read access for other users.** This setting permits all other users to read but not write to the OLAP model or metaoutline; that is, other users cannot save changes to the OLAP model or metaoutline.

- **Disallow all access for other users.** This setting denies all other users read or write permission to the OLAP model or metaoutline. You are the only user who can read or write to it.

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

- Change the OLAP model properties in Integration Services Console. See the Integration Services Console Help.

- Save the metaoutline with a different name by selecting File > Save As. See the Integration Services Console Help.

- To edit tables containing the permissions for the OLAP model or metaoutline, use the tools provided with the data source that contains OLAP Metadata Catalog. See “Viewing Permissions for OLAP Models” on page 35 or “Viewing Permissions for Metaoutlines” on page 36.

### Viewing Permissions for OLAP Models

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

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

- **MODEL_ID**: a system-generated reference number.

- **MODEL_NAME**: the name of the OLAP model; for example, TBC Model.
• MODEL_DESCRIPTION: a description of the OLAP model. If you do not enter a description when you save the model, this column is blank.

• MODEL_DATA_SOURCE: the name of the Open Database Connectivity (ODBC) data source on which the OLAP model is based; for example, TBC.

• MODEL_OWNER: the login name of the OLAP model owner; for example, sa. The login name is specified in the data source that contains the OLAP Metadata Catalog.

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

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

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

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

2 View the results.

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

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

Viewing Permissions for Metaoutlines

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

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

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

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

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

   ```sql
   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).

<table>
<thead>
<tr>
<th>MO_NAME</th>
<th>MO_OWNER</th>
<th>MO_ACCESS_CODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBC Metaoutline</td>
<td>sa</td>
<td>2</td>
</tr>
<tr>
<td>TBC_Archive</td>
<td>sa</td>
<td>1</td>
</tr>
<tr>
<td>TBC_Mine</td>
<td>sa</td>
<td>0</td>
</tr>
</tbody>
</table>

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

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

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

**Note:**

If you are accessing SAP BW data sources, the latest troubleshooting information can be found in the Hyperion® System™ 9 BI+™ Analytic Integration Connector for SAP BW™ readme file.

### Common Connection Problems

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

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

If you create an OLAP Metadata Catalog when logged in as one user name, you cannot access the tables in OLAP Metadata Catalog using a different user name unless you create an alias for the user name (for Microsoft SQL Server) or synonyms for the tables (for DB2 and Oracle). See the *Essbase Integration Services Installation Guide*.

- Are all required components up and running?
  - Essbase Integration Server
  - The database servers that manage OLAP Metadata Catalog and the data source databases
  - The data source database listener for OLAP Metadata Catalog and the data source

- Is the database client software installed and configured on the computer where Integration Server is installed?
  - Do the database client software utility programs, such as the command-line SQL utility, run? Can you connect to databases with them?
  - Does the user who starts Integration Server have the correct environment variables set and the necessary access privileges to run the database client software?
  - Are the required environment variables for the database client software, such as the path (and the library path on UNIX), configured on the computer where Integration Server is installed?

On UNIX, use `ivtestlib` located in the `ISHOME/bin` directory to confirm that the library path contains all files required by the open database connectivity (ODBC) driver.

- If necessary, is the `bin` directory for each database in the operating system path of the user who is running Integration Server?
- Does the ODBC data source name (DSN) configuration identify the name of the data source client?

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

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

Wait and retry the action when the data source is less busy. To avoid the problem, increase the ODBC driver time-out limit using ODBC Administrator on Windows systems. See the ODBC documentation.

- Did you create tables in OLAP Metadata Catalog? For information on creating tables for the catalog, see the *Essbase Integration Services Installation Guide*. 
Are you having problems related to ODBC while you are creating an Essbase outline? If so, turn on tracing, perform the action again, and analyze the results in the tracing log file. See “Using ODBC Tracing on Windows Systems” on page 61.

If you experience problems connecting to the data source, view the log for the data source for information about why the connection failed. See your data source documentation.

If none of the above steps help, review the list of database-specific problems in “Database-Specific Problems” on page 41.

**Database-Specific Problems**

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

- **DB2**: You may be required to bind the database driver to a DB2 server that is managing the databases you want. See the DB2 installation documentation.

- **Oracle OCI**: If connection attempts using Oracle OCI fail, be sure that when logged in as the Integration Services user (the user who installed Integration Services) on the Integration Server, you can successfully connect to the Oracle database using SQL*Plus. The name for the database that works in SQL*Plus should work in Integration Services when referred to as Oracle:Net Service Name (if you are using tnsnames) or Oracle:oname (if you are using Oracle onames).

- **SQL Server**: If you try to access an SQL Server database with the Microsoft-supplied ODBC driver when you do not have access permission, SQL Server connects you to the default database and does not notify you. Configure the ODBC DSN to use the database that you want to access. See the SQL Server documentation.

If none of the previous suggestions help, proceed to “ODBC and Connection Problems” on page 41 for an overview of connection problems, then see “Isolating Connection Problems” on page 42.

**ODBC and Connection Problems**

Integration Services Console must pass through three layers of software components to contact a data source: Integration Server, ODBC, and the database client software. Failure within or between any of these components can cause a lack of communication between Integration Services Console and the data source.

In some cases, the error messages received in Integration Services Console may not contain sufficient information for you to diagnose and resolve the problem, and you must go to the Integration Server to get more information and find a resolution.

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

- Software component problems:
- The components of Integration Services or the data source are not installed or are installed on the wrong computer.
- The components do not run because they are not set up properly.
- The components stopped running or were never started.
- The components are not compatible with the operating system.
- The versions of the different components are not compatible with each other.

- Configuration and communication problems:
  - The software components are not properly configured to communicate with each other.
  - User names, passwords, and permissions are configured or used inconsistently across the components and computers.
  - The communication interfaces of the software components are incompatible with each other.
  - Changes to components, databases, users, or permissions have put the software components out of sync with each other.

- Network and security problems:
  - A physical network path does not exist between the computers that must communicate with each other.
  - One or both computers attempting to communicate do not have the TCP/IP network protocol installed or configured.
  - The computer attempting to contact another computer does not have access to addressing information, such as a domain name server (DNS).
  - Security systems prevent all communication or certain types of communication between the computers that must talk with each other. For example, this is the case when client and server computers are separated by a firewall.

**Isolating Connection Problems**

If the list under “Common Connection Problems” on page 39 and your own knowledge fail to diagnose the connection problem, the recommended strategy is to test the components and connections starting at the data source and backtracking through the connections to Integration Services Console.

The client software must pass through three layers of components to connect to the data source containing OLAP Metadata Catalog and the data source, as illustrated in Figure 2.
The numbers in Figure 2 identify the general diagnostic steps that you should take to isolate connection problems. Follow these steps as indicated, starting with the connection between the database client software and the data source (1) and working backwards through the system to isolate connection problems. After you isolate a problem, refer to the appropriate components topic in “Correcting Connection Problems” on page 54.

To isolate connection problems:

1. Attempt to connect to the data source databases using a database client utility program. See “Testing the Database Client Connection to the Data Source” on page 44.

2. Attempt to connect to the data source databases using ODBC. See “Testing the ODBC Connection to the Data Source” on page 45.

3. Attempt to connect to the data source databases using Integration Server. See “Testing the Integration Server Connection to the Data Source” on page 49.
4 Attempt to connect to the data source databases using Integration Services Console. See “Testing the Integration Services Console Connection to Essbase Server” on page 53.

5 Attempt to connect to Essbase Server using Integration Server. See “Testing the Integration Server Connection to Essbase Server” on page 52.

6 Attempt to connect to Essbase Server using Integration Services Console. See “Testing the Integration Server Connection to Essbase Server” on page 52.

Testing the Database Client Connection to the Data Source

If your ODBC connections require database client software, the first step toward isolating connection problems in Integration Services is to attempt to connect the database client software to the data source that you are using. After you have verified that these components are communicating properly, you have a good base from which to test the remainder of the connection chain.

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

To test the database client connection to the data source:

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

Note:

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

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

   If the database client utility does not start, check to make sure that the client software is installed and the required environment variables are set. See “Correcting Database Client Problems” on page 56.

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

   If the connection fails, try to connect as a user who you know has access permission, such as an administrative user. If the administrative connection succeeds, check the other user accounts and permissions in the data source for the OLAP Metadata Catalog database, and carefully check user accounts with connection problems. See, “Correcting Data Source Problems” on page 56.

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

   \[ \text{SELECT } * \text{ FROM TBC.OV_INFO} \]

   If the select statement fails, check the permissions for the user name you are using to connect to the data source. See “Correcting Data Source Problems” on page 56. Also, check that the version
of the database client software you are using is compatible with the version of the data source. See “Correcting Database Client Problems” on page 56.

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

See “Testing the Integration Services Console Connection to the Data Source” on page 51. If the connection fails, proceed to “Testing the ODBC Connection to the Data Source” on page 45.

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

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

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

Testing the ODBC Connection to the Data Source

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

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

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

ODBC tracing can also be used to track down connection problems. See “Using ODBC Tracing” on page 61.

Testing ODBC on Windows Systems

➤ To test ODBC connections to the data source databases:

1 On the Windows desktop, select Start > Settings > Control Panel to open the Control Panel window.

2 In the Control Panel window, double-click the Administrative Tools icon, and then double-click the Data Sources (ODBC) icon.

3 In the ODBC Data Source Administrator dialog box, click the System DSN tab.

4 In the System Data Sources list, select the data source created for OLAP Metadata Catalog or the data source database, and click Configure.
A data source configuration dialog box is displayed. Depending on the data source and the driver that you are using, you may be asked to log on to the database immediately, or you may be asked to review all the settings before testing the connection. In either case, after attempting to log on to the database, observe the resulting message boxes to determine if you connected to the data source successfully.

**Note:**

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

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

See “Correcting ODBC Problems” on page 58.

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

See “Testing the Integration Services Console Connection to the Data Source” on page 51.

If the connection from Integration Services Console fails, proceed to “Testing the Integration Server Connection to the Data Source” on page 49.

### Testing ODBC on UNIX Systems

On UNIX systems, you use the `odbcconfig` utility to verify, add, or delete ODBC connections. You can use `odbcconfig` to perform preliminary testing. You must inspect the configuration files for ODBC manually or use a data source utility to perform more thorough testing. These configuration files include the login scripts that set the environment variables for ODBC and the `odbc.ini` file. See the documentation for the data source or the ODBC driver.

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

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

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

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

2 **Start the `odbcconfig` utility.**

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

   To edit and test the ODBC data sources, the available options are: List, Add, Delete, Test, and Exit.

3 **To list available DSNs, type 1 and press Enter.**

   Verify that the required DSN exists.

   If necessary, proceed to Step 4 to add the DSN.
4 If the required DSN does not exist, add it by completing the following steps:

Type 2 and press Enter.
At the prompts, type a database type, data source name, and a description.
You may also be prompted for additional information depending on the data source platform of the DSN that you are adding, such as server name, host name, or database name.
To verify the addition of the DSN, type 1 and press Enter. Verify that the DSN you just added is included in the list.

5 Type 5 and press Enter to exit the utility.

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

➢ To test an ODBC connection by using the odbcconfig utility:
1 Log on to the computer that runs Integration Server as the user who starts the Integration Server software.
2 Start the odbcconfig utility.
The utility lists the available options: List, Add, Delete, Test, and Exit.
3 To test the ODBC connection, type 4 and press Enter.
4 At the prompts, type a data source name, user name, and password.
   If no errors are returned, the connection is valid.
   In the event of errors, proceed to Step 5 to add corrected DSN information.
5 To add a DSN, type 2 at the starting odbcconfig utility window.
The odbcconfig utility does not permit editing of DSN information. You must add new DSN entries and delete any incorrect entries.
   At the prompts, type a data source name, user ID, and password.
6 Type 1 and press Enter to list DSNs and verify that the DSN you just added is included in the list.
7 Type 4 and press Enter to test the new ODBC connection.
   If no errors are returned, the connection is valid.
8 Repeat the steps in this section until the connection tests return no errors.
9 Type 5 and press Enter to exit the utility.

➢ To inspect the login script files:
1 Log on to the computer that runs Integration Server as the user who starts the Integration Server software.
2 In the current home directory, find the main login script file (typically, .profile for Korn Shell and Bourne Shell users, .login for C Shell users) and open it in a text editor.
Note:
The primary login script file may be named differently, depending on the UNIX operating system and the system administrator.

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

If one of these scripts is included in the main login script, make sure that the inclusion of the script file name follows the syntax for executing a script file within another script, and that the script file is specified with a complete directory path. See “Correcting ODBC Problems” on page 58.

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

4 After completing the preceding steps, test the connection by following the steps listed in “Testing the Integration Services Console Connection to Essbase Server” on page 53.

If you cannot make a connection, proceed to “Testing the Integration Server Connection to the Data Source” on page 49.

➤ To inspect the odbc.ini file:

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

2 To determine the location of the odbc.ini file, type echo $ODBCINI

   If this command does not display a file name and directory location, then you have not included the Integration Services script in the login script for the user. See “Correcting Integration Server Problems” on page 59.

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

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

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

   [ODBC Data Sources]
   TBC_MD=Essbase Integration Services sample catalog
   TBC=Essbase Integration Services sample database

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

   For Oracle:

   [myoracle]
   Driver=
   $HYPERION_Home/common/ODBC/Merant/5.0/Drivers/ARor819.so
   Description=my oracle source
   ServerName=mytnsServerName
For DB2:

[TBC_MD]
Driver=
   $HYPERION_Home/common/ODBC/Merant/5.0/Drivers/ARdb219.so
Description=DB2 database for sample testing

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

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

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

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

Tip:

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

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

See “Correcting ODBC Problems” on page 58.

**Testing the Integration Server Connection to the Data Source**

After verifying the connections between the ODBC, the database client, and the data source, the
next step in isolating a connection problem is to connect to the data source databases from
Integration Server. Test this connection by running Integration Services Shell on the same
computer that runs Integration Server. See Chapter 5, “Using Integration Services Shell.”

Note:

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

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

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

In this test, you set up these connections one at a time and execute a LOADMEMBER command
each time. By examining the errors each time, you can determine which connections are
successful.
To test the Integration Server connection to the data source:

1 **Verify that Integration Server is running.**

   If Integration Server is not running, start it. If Integration Server will not start, check that the Integration Services environment variables are set. See “Correcting Integration Server Problems” on page 59.

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

   olapid

   If Integration Services Shell does not start, make sure that the software is installed and that the Integration Services environment variables are set. See “Correcting Integration Server Problems” on page 59.

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

   login cypress

4 **Check the release of Integration Services by typing**

   version

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

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

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

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

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

   setcatalog "DSN=TBC_MD;CODEPAGE=English_UnitedStates.Latin1@Binary;UID=TBC;PWD=password"

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

7 **Set the instance of Essbase Server to which you want to connect by typing**

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

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

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

9 **Check the status of the preceding command by typing**

   status

   During a member load, Integration Services Shell connects first to the data source and then to OLAP Metadata Catalog. If the connection to the catalog is unsuccessful, the message “IS Error: Unable to log in to Catalog” is displayed. If the connection to the data source fails, the message “IS Error: Unable to login to Datasource” is displayed.
If the connection to the data source failed, verify the ODBC data source name, user name, and password. If the connection continues to fail, see “Correcting Integration Server Problems” on page 59 and “Correcting ODBC Problems” on page 58. Repeat Step 4 through Step 8 of this procedure to retest the connection. When you successfully connect to the data source, continue to the next step.

10 Test the connection to the OLAP Metadata Catalog database by attempting to perform a member load. Type:
   loadmember "APP=OLAP_TBC;DBN=Basic;OTL=TBC Metaoutline;"

11 Check the status of the preceding command by typing
   status

   During a member load, Integration Services Shell connects to OLAP Metadata Catalog and then to the 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 Integration Server Problems” on page 59 and “Correcting ODBC Problems” on page 58. Repeat Step 9 and Step 10 of this procedure to retest the connection. When you successfully connect to OLAP Metadata Catalog, continue to the next step.

12 Test the connection from Integration Services Console to the data source by proceeding to “Testing the Integration Services Console Connection to the Data Source” on page 51.

---

**Testing the Integration Services Console Connection to the Data Source**

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

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

1 **Verify that Integration Server is running.**

   If Integration Server is not running, start it. If Integration Server will not start, check that the Integration Services environment variables are set. See “Correcting Integration Server Problems” on page 59.

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

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

   If the connection fails, verify that you have specified the correct ODBC data source name for the OLAP Metadata Catalog database and the correct Integration Server computer name. Also verify that you specified the correct user name and password for the OLAP Metadata Catalog database.
database on the data source. See “Correcting Integration Services Console Problems” on page 60.

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

If the connection fails, verify that you have specified the correct ODBC data source name for the data source and the correct Integration Server computer name. Also verify that you specified the correct user name and password for the data source database on the data source. See “Correcting Integration Services Console Problems” on page 60.

Testing the Integration Server Connection to Essbase Server

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

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

1 Verify that Integration Server is running.

   If Integration Server is not running, start it. If Integration Server will not start, check to ensure that the Integration Services environment variables are set. See “Correcting Integration Server Problems” on page 59.

2 Verify that the Essbase Server is running.

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

   olapicmd

   If Integration Services Shell does not start, make sure that the software is installed and that the Integration Services environment variables are set. See “Correcting Integration Server Problems” on page 59.

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

   login cypress

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

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

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

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

   setcatalog "DSN=TBC_MD;CODEPAGE=English_UnitedStates.Latin1@Binary;UID=TBC;PWD=password"

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

7 Set the instance of Essbase Server to which you want to connect by typing
settarget "DSN=sequoia;UID=TBC;PWD=password;"

**Note:**

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

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

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

9 **Check the status of the preceding command by typing**

    status

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

After you can connect successfully to the Essbase Server, proceed to “Testing the Integration Services Console Connection to Essbase Server” on page 53.

**Testing the Integration Services Console Connection to Essbase Server**

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

**Note:**

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

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

1 **Verify that Integration Server is running.**
2 **Verify that Essbase Server is running.**
3 **Start Integration Services Console.**
4 **From Integration Services Console, connect to the Integration Server computer and OLAP Metadata Catalog; for example, cypress and TBC_MD.**
5 **After connecting to OLAP Metadata Catalog, open a metaoutline and connect to the data source for the metaoutline; for example, TBC Metaoutline and TBC.**
6 **Start a member load by selecting Outline > Member Load.**
If prompted, provide the name of the 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 Essbase Server and the correct user name and password. See the Essbase documentation.

If the 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.”

Note:

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

Correcting Connection Problems

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

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

- Are all user names and passwords being used consistently across all the components and computer user accounts? See “User Name and Password Problems” on page 55.
- Do all user accounts used in the system have the appropriate permissions and privileges?
- System administrators typically use highly privileged administrative accounts when setting up and testing a system. After the system is set up, users with fewer privileges may be unable to use the system successfully. Check permissions on the following components:
  - User account on the client computer
  - User account on the Integration Server computer
  - OLAP Metadata Catalog database
  - Data source database
  - User account for the 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 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 must have permissions to run all the following software components:

- Integration Server
- ODBC software
- Database client software
- Data source server (this may not be required)

**User Name and Password Problems**

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

**Tip:**

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

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

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

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

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

Adding aliases or synonyms to the data source databases adds to the number of user name and password sets you must manage. See the *Essbase Integration Services Installation Guide*. 
Tip:
During installation and connection testing, use a single set of user names and passwords to avoid confusion. Add user aliases or synonyms after the system is configured and running.

Correcting Data Source Problems

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

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

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

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

You can create the OLAP Metadata Catalog tables either automatically or by using the creation scripts provided with Integration Services. See the *Essbase Integration Services Installation Guide*.

- What are the exact names of the OLAP Metadata Catalog and data source databases? Are the names case-sensitive?

- Does the user name that you are using to connect to the data source have adequate permissions to access the database? Does the user have read and write permissions?

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

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

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

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

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

Correcting Database Client Problems

If you have problems connecting to the data source from the database client software, investigate the following possible causes:
● Is the database client software installed? Are all database client components required for the ODBC connections installed?

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

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

Earlier or later versions of the database client software may not work with the data source version that you are running. See your data source documentation.

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

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

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

● Are the data source server names, database names, user names, and passwords specified correctly?

Make sure you are using the correct names, including the exact spelling, uppercase or lowercase letters, and any required data source-specific identification syntax. See “Correcting Data Source Problems” on page 56.

● 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.

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. See the DB2 installation documentation.

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

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

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

* AIX and HP-UX: onsoctcp
* Solaris: ontlitcp

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

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

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

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

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

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

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

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

  **DB2:** On Windows systems, the DB2 Client Configuration Assistant shows the ODBC DSN. See “Correcting Data Source Problems” on page 56.

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

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

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

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

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

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

Most data source vendors provide utilities to test ODBC connections. For information, see the documentation for the data source.

If the preceding checks do not enable you to correct the problems connecting from ODBC to the data source, try using ODBC tracing to isolate the problem. See “Using ODBC Tracing” on page 61.

**Correcting Integration Server Problems**

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

- Is Integration Server installed? Are all the required Integration Server components installed?
  
  The following components must be installed to have a functional instance of Integration Server:
  
  - Integration Server software
  - Related base components (on UNIX, these components are included as part of Integration Server software)
  - DataDirect ODBC drivers (unless ODBC and drivers are already installed)

- Is Integration Server running? Is it configured properly?

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

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

  - Are the required environment variables set? Are they set for the correct user name?
    
    On Windows systems, if you decide not to enable the setup program to update environment variables automatically, you must update the variables manually.

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

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

  - Is port 3388 in use?
    
    Integration Server uses port 3388 to communicate with the Integration Services client software. If this port is being used by another program, Integration Server does not start.

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

    `ps -fe | grep olapisvr`
Many of the problems related to Integration Server not working properly may be traced to manual configuration steps that you may have missed. See the *Essbase Integration Services Installation Guide*.

**Correcting Integration Services Console Problems**

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

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

- Is Integration Services Console installed? Are all required components installed?
  The following components must be installed to have a functional Integration Services Console client:
    - Integration Services Console software
    - Related base components

- Is the Integration Services Console configured properly?
  Make sure the required environment variables are set. If you decide not to have the setup program update the environment variables, you must update them manually. See the *Essbase Integration Services Installation Guide*.

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

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

  See “Correcting Data Source Problems” on page 56.

- Does the user account with which you are running Integration Services Console have permission to connect to the computer running Integration Server?
  You must have a user account and appropriate access permissions on the server computer running Integration Server.

- If you have Integration Server installed on more than one computer, have you specified the correct server name?
  A particular Integration Server may not be set up to access the same OLAP Metadata Catalogs and data sources used by other servers.
Are you using adequate addressing to identify the Integration Server computer?

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

Using ODBC Tracing

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

Using ODBC Tracing on Windows Systems

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

➢ To use the tracing utility in ODBC Administrator:

1. Start ODBC Administrator on the computer running Integration Server.
2. Click the Tracing tab, specify a log file for the tracing utility, then click Start Tracing Now.

Note:

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

3. Start Integration Server and Integration Services Console.
4. From Integration Services Console, select Connections > OLAP Metadata Catalog > Connect, and try to connect to OLAP Metadata Catalog.
5. If you can connect to OLAP Metadata Catalog, open an OLAP model and try to connect to a data source.
6. If you cannot connect to OLAP Metadata Catalog (step 4) or to a data source (step 5), see the file that you specified as the log file in the root directory.

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

Using ODBC Tracing on UNIX Systems

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

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

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

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

**Note:**

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

4. Start Integration Server and Integration Services Console.

5. From Integration Services Console, select **Connections > OLAP Metadata Catalog > Connect**, and try to connect to OLAP Metadata Catalog.

6. If you can connect to OLAP Metadata Catalog, open an OLAP model and try to connect to a data source.

7. If you cannot connect to OLAP Metadata Catalog (step 5) or to a data source (step 6), read the `odbctrace.out` file.

   If necessary, share the information in the log file with Hyperion Technical Support to help solve the connectivity problem.
Tables in OLAP Metadata Catalog

In This Chapter

- OLAP Metadata Catalog ................................................................. 63
- Tables Relating to the OLAP Model ............................................... 64
- Tables Relating to the Metaoutline ............................................... 65
- Tables Relating to Drill-Through .................................................. 67
- Tables Relating to Hybrid Analysis .............................................. 67
- Miscellaneous Tables ................................................................. 68

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

This chapter describes the tables in OLAP Metadata Catalog.

**OLAP Metadata Catalog**

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

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

Regularly back up the database that contains OLAP Metadata Catalog so that you can restore OLAP Metadata Catalog from backups if it becomes corrupted. Use one of the procedures in Table 2, depending on what OLAP Metadata Catalog component you want to restore.

**Caution!**

It is strongly recommended that you do not alter the table names or data in OLAP Metadata Catalog with tools other than Integration Services.
Table 2  OLAP Metadata Catalog Restore Procedures

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

Tables Relating to the OLAP Model

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

- “Tables Relating to the Metaoutline” on page 65
- “Tables Relating to Drill-Through” on page 67
- “Tables Relating to Hybrid Analysis” on page 67
- “Miscellaneous Tables” on page 68

Table 3  OLAP Model Tables

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MB_FILTER</td>
<td>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.</td>
</tr>
<tr>
<td>MB_TX_RULE</td>
<td>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.</td>
</tr>
<tr>
<td>OA_INFO</td>
<td>Contains information about columns in a dimension table or a fact table. This information includes the column name, the OLAP model it belongs to, the dimension or fact table it belongs to, the source table and column on which it is based, whether it is hidden in OLAP models and metaoutlines, whether it is used for Drill-Through, whether it is enabled for attributes, and how to create it if it is a user-defined column.</td>
</tr>
<tr>
<td>OD_DETAILS</td>
<td>Contains information about dimensions in the OLAP model. This information includes system-generated ID numbers for parts of the OLAP model.</td>
</tr>
<tr>
<td>OD_INFO</td>
<td>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.</td>
</tr>
<tr>
<td>OM_INFO</td>
<td>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.</td>
</tr>
<tr>
<td>OM_PROPERTIES</td>
<td>Contains property information for the OLAP model.</td>
</tr>
</tbody>
</table>
### Table 4 Metaoutline Tables

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OMB_DETAILS</td>
<td>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.</td>
</tr>
<tr>
<td>OMB_INFO</td>
<td>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.</td>
</tr>
<tr>
<td>OV_INFO</td>
<td>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.</td>
</tr>
<tr>
<td>OVL_REL_DETAILS</td>
<td>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.</td>
</tr>
<tr>
<td>OVL_REL_INFO</td>
<td>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.</td>
</tr>
<tr>
<td>OVP_REL_DETAILS</td>
<td>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.</td>
</tr>
<tr>
<td>OVP_REL_INFO</td>
<td>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.</td>
</tr>
<tr>
<td>OM_DESCRIPTIONS</td>
<td>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.</td>
</tr>
<tr>
<td>JOIN_HINTS</td>
<td>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.</td>
</tr>
</tbody>
</table>

### Tables Relating to the Metaoutline

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

- “Tables Relating to the OLAP Model” on page 64
- “Tables Relating to Drill-Through” on page 67
- “Tables Relating to Hybrid Analysis” on page 67
- “Miscellaneous Tables” on page 68
<table>
<thead>
<tr>
<th>Table Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MO_INFO</td>
<td>Contains high-level information about a metaoutline: the name of the metaoutline, its description, the date it was last changed, the OLAP model on which the metaoutline is based, its owner, its access privileges, and whether the spreadsheet user of Drill-Through reports must enter the Integration Server name and a user name and password for the external data source.</td>
</tr>
<tr>
<td>MO_PROPERTIES</td>
<td>Contains information about metaoutline properties. This information includes the name and value of metaoutline properties, the name of the source table and column that Integration Services uses to assign the properties, the numeric range that represents a range specified by the user, Global Settings properties, and sort order. Sort order is based on a column in the data source, including columns not present in the OLAP model or metaoutline. Sort order can be ascending or descending. Also contains information on association of attributes (by level or by name) to base dimensions.</td>
</tr>
<tr>
<td>MOD_ATTRIBUTES</td>
<td>Contains information about attribute dimensions, such as attribute dimension type, alias names, and transformation rules.</td>
</tr>
<tr>
<td>MOD_INFO</td>
<td>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.</td>
</tr>
<tr>
<td>MS_INFO</td>
<td>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).</td>
</tr>
<tr>
<td>MS_PREDICATES</td>
<td>Contains information about members in a metaoutline. This information includes member filters.</td>
</tr>
<tr>
<td>MS_TX_RULE</td>
<td>Contains information about members in a metaoutline. This information includes member transformation rules.</td>
</tr>
<tr>
<td>MSR_INFO</td>
<td>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.</td>
</tr>
<tr>
<td>MSR_PREDICATES</td>
<td>Contains information about database measures in a metaoutline. This information includes measure filters.</td>
</tr>
<tr>
<td>MSR_TX_RULE</td>
<td>Contains information about database measures in a metaoutline. This information includes the name and definition of measure transformation rules.</td>
</tr>
<tr>
<td>MO_DESCRIPTIONS</td>
<td>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.</td>
</tr>
</tbody>
</table>
Tables Relating to Drill-Through

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

Table 5 describes each table in OLAP Metadata Catalog that stores information about drill-through reports. For information about the other tables in OLAP Metadata Catalog, see:

- “Tables Relating to the OLAP Model” on page 64
- “Tables Relating to the Metaoutline” on page 65
- “Tables Relating to Hybrid Analysis” on page 67
- “Miscellaneous Tables” on page 68

Table 5  Drill-Through Tables

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTR_ATTRIBUTES</td>
<td>Contains information about drill-through columns. This information includes the order in which Essbase Spreadsheet Add-in sorts and displays drill-through information.</td>
</tr>
<tr>
<td>DTR_INFO</td>
<td>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.</td>
</tr>
<tr>
<td>MO_INTERSECTION</td>
<td>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.</td>
</tr>
<tr>
<td>MO_PROPERTIES</td>
<td>Contains information about drill-through filters. This information includes the Essbase user name and password, drill-through report name, and metaoutline name.</td>
</tr>
</tbody>
</table>

Tables Relating to Hybrid Analysis

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

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

- “Tables Relating to the OLAP Model” on page 64
- “Tables Relating to the Metaoutline” on page 65
- “Tables Relating to Drill-Through” on page 67
- “Miscellaneous Tables” on page 68
### Hybrid Analysis Tables

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MO_PROPERTIES</td>
<td>Contains information about Hybrid Analysis columns. This information includes system-generated ID numbers for the OLAP model; the Hybrid Analysis column; and column precision, scale, and nullability.</td>
</tr>
</tbody>
</table>

### Miscellaneous Tables

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

- “Tables Relating to the OLAP Model” on page 64
- “Tables Relating to the Metaoutline” on page 65
- “Tables Relating to Drill-Through” on page 67
- “Miscellaneous Tables” on page 68

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CB_CONCURRENCY</td>
<td>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.</td>
</tr>
<tr>
<td>CB_VERSION</td>
<td>Contains information about the Integration Services software you have installed. This information includes the release number of Integration Services.</td>
</tr>
</tbody>
</table>
This chapter describes the commands supported by Essbase Integration Services Shell, the command-line tool provided with Integration Services. The majority of these commands provide the same functionality available through Essbase Integration Services Console. You use Integration Services Shell commands and scripts to create Essbase outlines and load Essbase databases.

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

**Starting Integration Services Shell**

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

- A data source database that contains the OLAP Metadata Catalog where the metadata is stored
- One or more data sources that you want to use to create OLAP models and metaoutlines
- Integration Services
  
  See “Starting Integration Server” on page 7.
- Essbase Server
  
  See the *Essbase Integration Services Installation Guide*. 
To start Integration Services Shell, use one of the following methods:

- From the Windows Start menu, select Programs > Hyperion > Integration Services > Integration Services Shell.
- Enter the executable name on the command line; for example, type `olapicmd`
- To set the TCP port number with which Essbase Integration Server communicates with its clients to a port different from the default 3388, enter the executable name and `-P portnumber` when you start Integration Services Shell; for example, type `olapicmd -P3390`

**Note:**

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

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

![Figure 3  Essbase Integration Services Shell Window](image)

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

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

**Integration Services Shell**

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

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

- Batch mode. To automate routine server maintenance and diagnostic tasks, write a script or batch file and run it from the command line.
Batch mode is convenient if you use a particular series of commands frequently or if the task requires many commands.

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

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

Tip:
It can be difficult to enter commands correctly when using the command line. The easiest method is to use the Integration Services Console OLAP Metaoutline standard user interface to perform a data or member load, and then choose to create a script file. You can then edit the script file as needed. Run all the commands in a script file by redirecting the standard input to use your command file. See “Running OLAP Command Script Files” on page 74.

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

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

SET commands provide Integration Server with the location of the source databases, 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:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SETSOURCE</td>
<td>Specifies the data source databases from which an Essbase outline is built</td>
</tr>
<tr>
<td>SETTARGET</td>
<td>Specifies the name of the Essbase Server computer on which an outline for an Essbase application and database is built</td>
</tr>
<tr>
<td>SETCATALOG</td>
<td>Specifies the OLAP Metadata Catalog that you created to store OLAP models and metaoutlines</td>
</tr>
<tr>
<td>SETLOCALE</td>
<td>Specifies the locale for OLAPICMD.</td>
</tr>
</tbody>
</table>

The STATUS command (see “STATUS” on page 76) 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:
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOADMEMBER</td>
<td>Loads members into an Essbase outline. You must load members before you load data.</td>
</tr>
<tr>
<td>LOADDATA</td>
<td>Loads data into an Essbase database. You cannot load data without loading members.</td>
</tr>
<tr>
<td>LOADALL</td>
<td>Loads both members and data into an Essbase database.</td>
</tr>
</tbody>
</table>

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 “Performing Member Loads Using Integration Services Shell” on page 88 for a sample sequence of commands for loading members.

See “Loading Data Using Integration Services Shell” on page 88 for a sample sequence of commands for loading data.

**Informational Commands**

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

**Command Syntax and Execution**

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

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

```plaintext
SETSOURCE "DSN=database;CODEPAGE=English_UnitedStates.Latin1@Binary;UID=Smith;PWD=password;"
```

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

In this guide, all optional syntax parameters are indicated by their enclosure in brackets [ ]; however, do not type the brackets when you enter the commands.

Press Enter to execute the commands.
Note:
If you are unsure of a specific syntax, use Integration Services Console to schedule member or data loads, or to create load scripts. An Integration Services Shell batch file and a corresponding .cbs file are created automatically in the ISHOME\Batch directory. You can reference, modify, and reuse the .cbs file for future member and data loads. In the Schedule Essbase Load dialog box, be sure to check the Save Script Only option instead of the Schedule option. See “Recording Member and Data Load Scripts” on page 73.

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

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

Running Integration Services Shell Script and Batch Files
If you use a series of commands frequently or you must enter many commands to complete a task, you can automate the task with a script or batch file. Both are text files.

● A script file (.cbs extension) contains Integration Services Shell commands. You can run a script file from the operating system command line or from within an operating system batch file.

● On Windows platforms, a batch file (.bat extension) is an operating system file that calls multiple Integration Services Shell scripts and is used to run multiple sessions of Integration Services Shell commands. You can run a batch file on the server from the operating system prompt.

● On UNIX, shell scripts are used in place of batch or script files.
When you run a script file, Integration Services Shell executes the commands in the order specified in the script until the program reaches the end of the file.

**Running OLAP Command Script Files**

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

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

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

For example, the following sample script file, `olap_tbc.cbs`, was created in a text editor. This script connects to Essbase from the Integration Server computer and generates outlines for a sample database. In the following example, the `status` command (see “STATUS” on page 76) is used to check the success or failure of each command. The `status` command returns the execution status in the Integration Services Shell window.

```
login labmachine1
status
settarget "DSN=labmachine1;UID=hyperion;PWD=password"
status
setcatalog *DSN=TBC_MD;CODEPAGE=
   English_UnitedStates.Latin1@Binary;UID=tbc;PWD=password"
status
setsource "DSN=TBC;CODEPAGE=
   English_UnitedStates.Latin1@Binary;UID=tbc;PWD=password"
status
loadall "OTL=TBC Metaoutline;APP=olaptbc;DBN=Tbf1;FLT_ID=1;
   CALC_SCRIPT=#DEFAULT#;"
status
loadall "OTL=TBC Metaoutline;APP=olaptbc;DBN=Tbf2;FLT_ID=2;
   OTL_CLEAR=Y;"
status
loadall "OTL=TBC Metaoutline;APP=olaptbc;DBN=Tbf3a;FLT_ID=3;
   OTL_CLEAR=N;CALC_SCRIPT=#DEFAULT#;"
status
loadall "OTL=TBC Metaoutline;APP=olaptbc;DBN=Tbf3ec;FLT_ID=3;
   OTL_CLEAR=N;ESSC_SCRIPT=mytest1"
status
exit
```

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

```
olapicmd -f olap_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 -f olap_tbc.cbs > olap_tbc.log
```
Integration Services Shell Commands

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

The following commands are described in the subtopics:

● “LOGIN” on page 75
● “STATUS” on page 76
● “VERSION” on page 76
● “SETLOCALE” on page 76
● “SETSOURCE” on page 77
● “SETTARGET” on page 78
● “SETCATALOG” on page 78
● “LOADMEMBER” on page 78
● “LOADDATA” on page 81
● “LOADALL” on page 83
● “LOGOUT” on page 87
● “SHUTDOWN” on page 87
● “EXIT” on page 88

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

The load commands in this topic are shown with an abbreviated syntax, which includes the syntax required for incremental loads. To see the full syntax for standard or incremental loads, use the Integration Services Console to record a script for a load. See “Recording Member and Data Load Scripts” on page 73.

LOGIN

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

Logging in to Integration Server Running on the Default Port

Syntax

LOGIN ISHost

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

Example
LOGIN cypress

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

Syntax

LOGIN IShost:non-default_port_number

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

non-default_port_number The nonstandard port number on which Integration Server listens

Example

LOGIN cypress:3390

STATUS

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

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

Syntax

STATUS

Example

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

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

VERSION

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

Syntax

VERSION

Example

VERSION returns the release number for Integration Services software.

SETLOCALE

The SETLOCALE command enables you to override the default locale for OLAPICMD. The default locale for OLAPICMD is UTF-8 (Unicode).

Although a script file saved from Integration Services Console does not have the UTF-8 signature, OLAPICMD nevertheless reads the script as UTF-8.
If you create a script file in Microsoft Notepad and save the script file as a UTF-8 file, you do not need to use the SETLOCALE command to select a locale. You can use the default locale. If you choose to select a locale, you must select one that is UTF-8. A native locale causes load operations to fail.

If you create a script file in Microsoft Notepad and save the script file as an ANSI file, you must select a native locale. A UTF-8 locale causes load operations to fail.

If you intend to enter Integration Services Shell commands manually (rather than executing them through a script) on a native computer, you must select the appropriate native locale using the SETLOCALE command. Failure to do so will cause load operations to fail.

**Syntax**

```
SETLOCALE <LANGUAGE_TERRITORY.CODEPAGE@SORT>
SETLOCALE .UTF8@default
```

**Examples**

```
SETLOCALE Japanese_Japan.MS932@Binary
SETLOCALE .UTF8@default
```

---

**SETSOURCE**

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

**Syntax**

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

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

**Example**

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

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

Syntax

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

*EssbaseServices dataservername*  The TCP/IP name or number of the computer where Essbase Server is running

*username*  The name for logging on to Essbase Server

*password*  The password for logging on to Essbase Server—case-sensitive

Example

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

SETCATALOG

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

Syntax

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

*ODBCdatasource*  The ODBC data source name configured on the computer where Integration Server runs—case-sensitive

*Codepage*  The code page of the language you want to use during the current Integration Services Console session

*username*  The name for logging on to the ODBC data source—case-sensitive

*password*  The password for logging on to the ODBC data source—case-sensitive

Example

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

LOADMEMBER

The LOADMEMBER command initiates a member load operation on Integration Server. The load environment must be set up properly before you invoke the LOADMEMBER command. To set up the load environment, issue the SET commands in any order. See “SET and LOAD Commands” on page 71.

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

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

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

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

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

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

This is the format:
DELETE_DB=[Y|N]

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

Values are as follows:
Y Delete and restore (re-create) the Essbase database during the
member load process.
N Do not delete and restore the Essbase database (this is the default).

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

This process is slower than creating or updating an Essbase outline
without deleting the members; therefore, do not delete all members first
unless it is necessary. You should delete all members if, for example,
you know that some members have been removed from the OLAP
metaoutline and you want to build an Essbase outline containing a
smaller set of members. The default is N (No).

This is the format:
OTL_CLEAR=[Y|N]

Values are as follows:
Remove all members from the existing database and then re-create the database by using the members in the metaoutline.

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

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

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

\[
\text{INCUPD} = \text{DimID-MbrID, DimID-MbrID, ... DimID-MbrID; [;]}
\]

Values for incremental update are as follows:

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

**Dynamic Restructuring Member Load Options**

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

Values are as follows:

1. **Preserve all data.** Select to preserve all existing data that applies to the changed outline when restructuring occurs. This is the default (INCUPD_MEMBER=1).
2. **Discard all data.** Select to clear all data from the database.
3. **Preserve level 0 data.** Select to preserve data only for level zero members.
   - This is the optimal restructure option if you change the source database and need to recalculate the data, and if all data required for the calculation is in level zero members.
   - Selecting this option deletes all upper-level blocks before restructuring. This reduces the disk space for restructuring and improves calculation time when the database is recalculated. The upper-level blocks are re-created when you calculate the database.
4. **Preserve input data.** Select to preserve only those blocks containing data that is loaded. Many applications contain data that is entered at parent levels. Selecting this option prevents deletion of any blocks that are created by data load, whether they are non-level zero or level zero (leaf member) blocks.

**Attribute Update Level**

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

Values are as follows:

0. **Delete all attribute dimensions after the member load is performed.**
1. **Do not update existing attribute dimensions while updating other types of dimensions (for example: standard, time, or measures dimensions) during a member load.**
2. **Update all existing attribute dimensions and add new attribute dimensions during a member load.**
Username List  
Optional. Lists all user IDs that can access the Essbase application and database into which you are loading the members. Commas separate user IDs. For example, @@USERS="admin","ljones","ksmith".

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

Create Unicode Application  
Optional. You can specify that a Unicode application is created on Essbase Server. The default is that no Unicode application is created. This parameter cannot be overwritten. For example, if a previous application exists and you are overwriting that application with the current member load, you cannot overwrite the original Unicode or non-Unicode setting.

This is the format:
UNICODE=[Y|N]

For example:
UNICODE=Y

Update Drill-Through Data Only  
Optional. Update drill-through information or hybrid analysis information.

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

This is the format:
UPDATE_DTDATA_ONLY=[Y|N]

For example:
UPDATE_DTDATA_ONLY=Y

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

LOADDATA
The LOADDATA command initiates a data load operation on Integration Server. The load environment must be set up properly before you invoke this command. To set up the load environment, issue the SET commands in any order. See “SET and LOAD Commands” on page 71.

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

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

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

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

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

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

This is the format:

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

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

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

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

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

Values for incremental update are as follows:

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

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

Values are as follows:

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

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

**Username List**

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

**Calc Script Name**

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

**Note:** If the data storage model property of the metaoutline is aggregate storage, this parameter is not applicable.

**Esscmd Script File**

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

**FactTable Column for Incremental Update**

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

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

This is the format:

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

For example:

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

**Example**

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

**LOADALL**

The LOADALL command initiates a load of both members and data on Integration Server. The load environment must be set up properly before you invoke this command. To set up the load environment, issue the SET commands in any order. See “SET and LOAD Commands” on page 71.

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

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

**Syntax**

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

<table>
<thead>
<tr>
<th><strong>Metaoutline</strong></th>
<th>The name of the metaoutline. Integration Server uses the specified metaoutline to extract data from the data source to create an Essbase outline.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Essbase Application</strong></td>
<td>The name of the Essbase application on the target Essbase Server computer where the Essbase outline is created.</td>
</tr>
<tr>
<td><strong>Essbase Database</strong></td>
<td>The name of the Essbase database on the target Essbase Server computer. Integration Server applies this name to the Essbase outline.</td>
</tr>
<tr>
<td><strong>Member Load Filter ID</strong></td>
<td>Optional. You can specify a member load filter ID to use when loading members. If you do not specify a filter ID, the default name (*DEFAULT) is used. The default filter ID is 1; any additional filters are numbered sequentially, starting with the number 2. For information about creating filters, see the Integration Services Console Help.</td>
</tr>
<tr>
<td><strong>Data Load Filter ID</strong></td>
<td>Optional. You can specify a data load filter ID to use when loading data. If you do not specify a filter ID, the default filter (*DEFAULT) is used. The default filter ID is 1; any additional filters are numbered sequentially, starting with the number 2. For information about creating filters, see the Integration Services Console Help.</td>
</tr>
<tr>
<td><strong>Delete and Restore Database</strong></td>
<td>Optional. The Delete and Restore Database parameter directs whether Integration Server should delete all members in the Essbase database before performing a member load. The default is N (No). This is the format: DELETE_DB=[Y</td>
</tr>
</tbody>
</table>
**Replace Zeros with #MISSING**

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

This is the format:

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

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

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

**Delete All Members First**

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

This process is slower than creating or updating an Essbase outline without deleting the members; therefore, do not delete all members first unless it is necessary. You should delete all members if, for example, you know that some members have been removed from the OLAP metaoutline and you want to build an Essbase outline containing a smaller set of members. The default is N (No).

This is the format:

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

Values are as follows:

- Y Remove all members from the existing database and then re-create the database by using the members in the metaoutline.
- N Do not remove members from the existing database; only update the Essbase database (this is the default).

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

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

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

Values for incremental update are as follows:

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

**Dynamic Restructuring Member Load Options**

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

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

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

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

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

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

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

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

Values are as follows:

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

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

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

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

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

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

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

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

**Calc Script Name**  
**Optional.** The name of the calculation script. The calculation script determines how Essbase calculates the data values in the database. If you do not select a calculation script, Integration Server does not perform a calculation. If the data storage model property of the metaoutline is aggregate storage, this parameter is not applicable.

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

**Create Unicode Application**

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

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

This is the format:

UNICODE=[Y|N]

For example:

UNICODE=Y

**Example**

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

**LOGOUT**

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

**Syntax**

LOGOUT

**Example**

LOGOUT logs you out of Integration Server.

**Note:**

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

**SHUTDOWN**

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

**Syntax**

SHUTDOWN

**Example**

SHUTDOWN shuts down Integration Server.
**EXIT**

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

**Syntax**

EXIT

**Example**

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

---

**Performing Member Loads Using Integration Services Shell**

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

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

➤ To load members and dimensions with Integration Services Shell, follow the same steps for loading members that you use with Integration Services Console:

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

2. **Connect to the external data source by using the SETSOURCE command; for example**
   
   SETSOURCE DSN=TBC;CODEPAGE=English_UnitedStates.Latin1@Binary;UID=TBC;PWD=password

3. **Connect to the 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;CODEPAGE=English_UnitedStates.Latin1@Binary;UID=TBC;PWD=password

5. **Start the member load by using the LOADMEMBER command; for example**
   
   LOADMEMBER "OTL=TBC Metaoutline;APP=OLAP_TBC;DBN=Basic;FLT_NAME=Filter1"

---

**Loading Data Using Integration Services Shell**

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

You can manually schedule the batch file by enabling the Task Scheduler service on Windows 2000, 2003 and Windows XP or the cron scheduling daemon on UNIX systems. If you are not sure how to schedule a batch file manually, see the operating system documentation.
To load data with Integration Services Shell, follow the same steps for loading data that you use with Integration Services Console:

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

2. Connect to the external data source by using the SETSOURCE command; for example
   
   ```
   SETSOURCE DSN=TBC;CODEPAGE=English_UnitedStates.Latin1@Binary;UID=TBC;PWD=password
   ```

3. Connect to the 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;CODEPAGE=English_UnitedStates.Latin1@Binary;UID=TBC;PWD=password
   ```

5. Start the data load by using the LOADDATA command; for example
   
   ```
   LOADDATA "OTL=TBC Metaoutline;APP=OLAP_TBC;DBN=Basic;FLT_NAME=Filter1"
   ```
This chapter describes the rules for naming applications, databases, dimensions, members, and aliases in Essbase. For detailed information on creating Essbase applications and databases, see the Essbase product documentation.

### Naming Restrictions for Applications and Databases

When naming applications and databases, follow these rules:

- Use no more than 8 bytes when naming non-Unicode-mode applications and databases; use no more than 30 characters when naming Unicode-mode applications and databases.
- Do not use spaces anywhere in the name.
- Do not use the following special characters anywhere in the name:

<table>
<thead>
<tr>
<th>Character</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>asterisk</td>
</tr>
<tr>
<td>+</td>
<td>plus</td>
</tr>
<tr>
<td>\</td>
<td>back slash</td>
</tr>
<tr>
<td>?</td>
<td>question mark</td>
</tr>
<tr>
<td>[</td>
<td>brackets</td>
</tr>
<tr>
<td>'</td>
<td>quotation marks</td>
</tr>
<tr>
<td>:</td>
<td>colon</td>
</tr>
<tr>
<td>;</td>
<td>semicolon</td>
</tr>
<tr>
<td>,</td>
<td>comma</td>
</tr>
<tr>
<td>‘</td>
<td>apostrophe</td>
</tr>
<tr>
<td>=</td>
<td>equals</td>
</tr>
<tr>
<td>/</td>
<td>forward slash</td>
</tr>
<tr>
<td>&gt;</td>
<td>greater than</td>
</tr>
<tr>
<td>&lt;</td>
<td>less than</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>.</td>
<td>period</td>
</tr>
</tbody>
</table>

- For aggregate storage databases only, do not use any of the following words as application or database names:
Enter the name in the case in which you want the word displayed. The application or database name is created exactly as you enter it. If you enter the name as all capital letters (for instance, NEWAPP), Essbase does not automatically convert it to upper- and lowercase (for instance, Newapp).

### Naming Restrictions for Dimensions, Members, and Aliases

When naming dimensions, members, and aliases in the database outline, follow these rules:

- Use no more than 80 characters when naming dimensions, members, or aliases.
- Names are not case-sensitive unless case-sensitivity is enabled. See “Setting Outline Properties” in *Essbase Administration Services Online Help*.
- Do not use quotation marks (" "), brackets ([ ]), or tabs anywhere in a name.

**Note:**

Brackets ([ ]) are supported in block storage outlines, but are not recommended because their use causes errors when converting to an aggregate storage outline.

- At the beginning of a dimension or member name, do not use the following characters:

<table>
<thead>
<tr>
<th>@ at</th>
<th>() parentheses</th>
</tr>
</thead>
<tbody>
<tr>
<td>\ back slash</td>
<td>. period</td>
</tr>
<tr>
<td>{} braces</td>
<td>+ plus</td>
</tr>
<tr>
<td>, comma</td>
<td>' apostrophe</td>
</tr>
<tr>
<td>- dash, hyphen, or minus</td>
<td>_ underscore</td>
</tr>
<tr>
<td>= equals</td>
<td></td>
</tr>
<tr>
<td>&lt; less than</td>
<td></td>
</tr>
</tbody>
</table>

- Do not place spaces at the beginning or end of a name. Essbase ignores spaces at the beginning or end of a name.
- Do not use the following words as dimension or member names:
  - Calculation script commands, operators, and keywords. For a list of commands, see the *Essbase Technical Reference*.
  - Report writer commands. For a list of commands, see the *Essbase Technical Reference*. 

---

92  Naming Restrictions for Essbase Applications, Databases, and Members
Function names and function arguments. For a list of functions, see the *Essbase Technical Reference*.

Names of other dimensions and members (unless the member is shared), and generation names, level names, and aliases in the database.

Any of the following words:

<table>
<thead>
<tr>
<th>ALL</th>
<th>GENRANGE</th>
<th>OR</th>
</tr>
</thead>
<tbody>
<tr>
<td>AND</td>
<td>GROUP</td>
<td>PAREN</td>
</tr>
<tr>
<td>ASSIGN</td>
<td>GT</td>
<td>PARENPARM</td>
</tr>
<tr>
<td>CALC</td>
<td>ID</td>
<td>PERCENT</td>
</tr>
<tr>
<td>CALCMBR</td>
<td>IDERROR</td>
<td>PLUS</td>
</tr>
<tr>
<td>COPYFORWARD</td>
<td>INTEGER</td>
<td>RELOP</td>
</tr>
<tr>
<td>CROSSDIM</td>
<td>LE</td>
<td>SET</td>
</tr>
<tr>
<td>CURMBRNAME</td>
<td>LEVELRANGE</td>
<td>SKIPBOTH</td>
</tr>
<tr>
<td>DIM</td>
<td>LOOPBLOCK</td>
<td>SKIPMISSING</td>
</tr>
<tr>
<td>DIMNAME</td>
<td>LOOPARMS</td>
<td>SKIPNONE</td>
</tr>
<tr>
<td>DIV</td>
<td>LT</td>
<td>SKIPZERO</td>
</tr>
<tr>
<td>DYNAMIC</td>
<td>MBR</td>
<td>TO</td>
</tr>
<tr>
<td>EMPTYPARAM</td>
<td>MBRNAME</td>
<td>TOLOCALRATE</td>
</tr>
<tr>
<td>EQ</td>
<td>MBRONLY</td>
<td>TRAILMISSING</td>
</tr>
<tr>
<td>EQOP</td>
<td>MINUS</td>
<td>TRAILSUM</td>
</tr>
<tr>
<td>EXCEPT</td>
<td>MISSING</td>
<td>UMINUS</td>
</tr>
<tr>
<td>EXP</td>
<td>MUL</td>
<td>UPPER</td>
</tr>
<tr>
<td>EXPERROR</td>
<td>MULOP</td>
<td>VARORXMBR</td>
</tr>
<tr>
<td>FLOAT</td>
<td>NE</td>
<td>XMBRONLY</td>
</tr>
<tr>
<td>FUNCTION</td>
<td>NON</td>
<td>$$$UNIVERSEX $$</td>
</tr>
<tr>
<td>GE</td>
<td>NONINPUT</td>
<td>#MISSING</td>
</tr>
<tr>
<td>GEN</td>
<td>NOT</td>
<td>#MI</td>
</tr>
</tbody>
</table>

**Note:**

If you enable Dynamic Time Series members, do not use the associated generation names, including History, Year, Season, Period, Quarter, Month, Week, or Day.
Using Dimension and Member Names in Calculation Scripts, Report Scripts, Formulas, and Filters

In calculation scripts, report scripts, filter definitions, partition definitions, or formulas, you must enclose member names in quotation marks (" ") for block storage databases, and in brackets ([ ]) for aggregate storage databases, in the following situations:

- The name starts with one or more numerals (for example, 100).
- The name contains spaces or any of the following characters:

<table>
<thead>
<tr>
<th>Character</th>
<th>Equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;</td>
<td>&amp;</td>
</tr>
<tr>
<td>&gt;</td>
<td>&gt;</td>
</tr>
<tr>
<td>*</td>
<td>&lt;</td>
</tr>
<tr>
<td>@</td>
<td>(</td>
</tr>
<tr>
<td>\</td>
<td>)</td>
</tr>
<tr>
<td>%</td>
<td>.</td>
</tr>
<tr>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>,</td>
<td>+</td>
</tr>
<tr>
<td>-</td>
<td>/</td>
</tr>
<tr>
<td>=</td>
<td>~</td>
</tr>
</tbody>
</table>

- In calculation scripts and formulas, you must enclose the following member names in quotation marks (" ") for block storage databases, and in brackets ([ ]) for aggregate storage databases:

<table>
<thead>
<tr>
<th>Member Name</th>
<th>Equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>BEGIN</td>
<td>MEMBER</td>
</tr>
<tr>
<td>DOUBLE</td>
<td>RANGE</td>
</tr>
<tr>
<td>END</td>
<td>STRING</td>
</tr>
<tr>
<td>MACRO</td>
<td>THEN</td>
</tr>
</tbody>
</table>
This appendix provides a partial list of the return codes and error messages generated by Essbase Integration Server and Essbase Server. Return codes and error messages are grouped into two categories: return codes and errors generated by Integration Server and errors generated by Essbase Server for member and data loads.

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

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

For errors generated by Essbase for member and data loads, see “Essbase Error Messages Generated During Data Loads” on page 103.

**OLAPICMD Session Return Codes**

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

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

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

- The `winrtcd.txt` file lists the return codes for Windows and the associated message text. The return code values in Windows are unique.

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

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

<table>
<thead>
<tr>
<th>UNIX Return Code</th>
<th>Windows Return Code</th>
<th>Message Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>061</td>
<td>1195069</td>
<td>Failed to add Essbase member <code>member_name</code> (DUPLICATE, ignored).</td>
</tr>
<tr>
<td>061</td>
<td>1195325</td>
<td>There must be an active session to process this command.</td>
</tr>
<tr>
<td>061</td>
<td>2003005</td>
<td>Data source error. Could not locate the metadata attributes.</td>
</tr>
<tr>
<td>069</td>
<td>1195077</td>
<td>Failed to create a local context.</td>
</tr>
<tr>
<td>069</td>
<td>1195333</td>
<td>ODBC Error. Encountered unknown ODBC exception while opening database.</td>
</tr>
<tr>
<td>094</td>
<td>1195358</td>
<td>Failed to get user attributes.</td>
</tr>
<tr>
<td>094</td>
<td>2002014</td>
<td>Cube builder error. Client request error. Invalid number of parameters.</td>
</tr>
</tbody>
</table>

**Member Load Error Messages**

Table 9 lists some of the error messages that Integration Server may generate during a member load. Italicized words represent variable names inserted in the error message. Refer to the error message number in the first column and make a note of it if you need to contact Hyperion Technical Support with a problem.

<table>
<thead>
<tr>
<th>Windows Message Number</th>
<th>Message Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>1195049</td>
<td>Detected metadata join problem (<code>table.column</code> &lt;-&gt; <code>table.column</code>).</td>
</tr>
<tr>
<td>1195050</td>
<td>Encountered unknown ODBC exception while opening database. Aborting the command.</td>
</tr>
<tr>
<td>1195064</td>
<td>Failed to add database <code>Essbase database</code>.</td>
</tr>
<tr>
<td>1195065</td>
<td>Failed to add dimension <code>dimension</code>.</td>
</tr>
<tr>
<td>Windows Message Number</td>
<td>Message Content</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>1195066</td>
<td>Failed to add Essbase member <code>member</code> (%d).</td>
</tr>
<tr>
<td>1195067</td>
<td>Failed to add Essbase member <code>member</code> (child of shared parent).</td>
</tr>
<tr>
<td>1195068</td>
<td>Failed to add Essbase member <code>member</code> (DUPLICATE).</td>
</tr>
<tr>
<td>1195069</td>
<td>Failed to add Essbase member <code>member</code> (DUPLICATE, ignored).</td>
</tr>
<tr>
<td>1195070</td>
<td>Failed to add Essbase member <code>member</code> (no parent or sibling members found).</td>
</tr>
<tr>
<td>1195071</td>
<td>Failed to add Essbase member <code>member</code>, %d.</td>
</tr>
<tr>
<td>1195072</td>
<td>Failed to add Essbase member <code>member</code>; the member exists.</td>
</tr>
<tr>
<td>1195073</td>
<td>Failed to add formula <code>formula</code>.</td>
</tr>
<tr>
<td>1195074</td>
<td>Failed to add member <code>member</code>.</td>
</tr>
<tr>
<td>1195075</td>
<td>Failed to add metaoutline member <code>member</code>.</td>
</tr>
<tr>
<td>1195076</td>
<td>Failed to add user attribute <code>user-defined attribute</code>.</td>
</tr>
<tr>
<td>1195077</td>
<td>Failed to create a local context.</td>
</tr>
<tr>
<td>1195078</td>
<td>Failed to create a local outline.</td>
</tr>
<tr>
<td>1195079</td>
<td>Failed to create application <code>Essbase application</code>.</td>
</tr>
<tr>
<td>1195080</td>
<td>Failed to create dimension <code>dimension</code>.</td>
</tr>
<tr>
<td>1195083</td>
<td>Failed to get information for parent of recursive hierarchy <code>member</code>.</td>
</tr>
<tr>
<td>1195085</td>
<td>Failed to get metadata for <code>dimension.member</code>.</td>
</tr>
<tr>
<td>1195088</td>
<td>Failed to initialize Essbase API.</td>
</tr>
<tr>
<td>1195089</td>
<td>Failed to open outline <code>Essbase outline</code>.</td>
</tr>
<tr>
<td>1195093</td>
<td>Failed to restructure <code>Essbase outline</code>.</td>
</tr>
<tr>
<td>1195095</td>
<td>Failed to update Essbase server (NULL HEADER).</td>
</tr>
<tr>
<td>1195097</td>
<td>Filter <code>metaoutline filter</code> doesn’t exist for <code>metaoutline metaoutline</code>.</td>
</tr>
<tr>
<td>1195098</td>
<td>Filter expression length exceeded the limit of maximum bytes bytes.</td>
</tr>
<tr>
<td>1195115</td>
<td>Incremental update specification <code>incremental update specification string</code> is incorrect.</td>
</tr>
<tr>
<td>1195126</td>
<td>Invalid aggregation function <code>aggregate function for table.column</code>.</td>
</tr>
<tr>
<td>1195129</td>
<td>Invalid esscmd script name <code>esscmd script</code>.</td>
</tr>
<tr>
<td>1195130</td>
<td>Invalid parameter <code>parameter</code>.</td>
</tr>
<tr>
<td>1195132</td>
<td>Invalid parameter string.</td>
</tr>
<tr>
<td>Windows Message Number</td>
<td>Message Content</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>1195142</td>
<td>Metadata error: <code>property ID</code> is not a valid property for <code>member</code> or <code>dimension</code>.</td>
</tr>
<tr>
<td>1195143</td>
<td>Metaoutline validation error. <code>Recursive hierarchy member</code>, a recursive member cannot have aggregation filter.</td>
</tr>
<tr>
<td>1195144</td>
<td>Metaoutline validation error. Filter <code>~ member filter</code>, is not associated with any member.</td>
</tr>
<tr>
<td>1195188</td>
<td>Unknown exception encountered while opening dimension <code>dimension</code>.</td>
</tr>
<tr>
<td>1195189</td>
<td>Unknown exception encountered while opening member <code>member</code>.</td>
</tr>
<tr>
<td>1195190</td>
<td>Unknown exception encountered while opening metaoutline <code>metaoutline</code>.</td>
</tr>
<tr>
<td>1195341</td>
<td>Failed to Login to Essbase Server.</td>
</tr>
<tr>
<td>1195342</td>
<td>Failed to Get Essbase Settings.</td>
</tr>
<tr>
<td>1195344</td>
<td>Failed to unload database.</td>
</tr>
<tr>
<td>1195345</td>
<td>Failed to delete database.</td>
</tr>
<tr>
<td>1195346</td>
<td>Failed to close outline.</td>
</tr>
<tr>
<td>1195347</td>
<td>Failed to read outline.</td>
</tr>
<tr>
<td>1195349</td>
<td>Failed to unlock Database <code>database</code>.</td>
</tr>
<tr>
<td>1195350</td>
<td>Failed to save outline.</td>
</tr>
<tr>
<td>1195351</td>
<td>Unable to process state for Essbase command.</td>
</tr>
<tr>
<td>1195352</td>
<td>Failed to get access permissions for application.</td>
</tr>
<tr>
<td>1195353</td>
<td>Failed to set access permissions for application.</td>
</tr>
<tr>
<td>1195354</td>
<td>Failed to get access permissions for database.</td>
</tr>
<tr>
<td>1195355</td>
<td>Failed to set access permissions for database.</td>
</tr>
<tr>
<td>1195356</td>
<td>No member found.</td>
</tr>
<tr>
<td>1195357</td>
<td>Essbase Error: <code>error definition</code>.</td>
</tr>
<tr>
<td>1195358</td>
<td>Failed to get user attributes.</td>
</tr>
<tr>
<td>1195360</td>
<td>Failed to add member to outline.</td>
</tr>
<tr>
<td>1195361</td>
<td>Failed to calculate the data.</td>
</tr>
<tr>
<td>1195362</td>
<td>Failed to get parent.</td>
</tr>
<tr>
<td>1195363</td>
<td>Failed to get member.</td>
</tr>
<tr>
<td>1195364</td>
<td>Failed to move member.</td>
</tr>
<tr>
<td>1195365</td>
<td>Failed to get shared member.</td>
</tr>
<tr>
<td>Windows Message Number</td>
<td>Message Content</td>
</tr>
<tr>
<td>------------------------</td>
<td>------------------------------------------------------</td>
</tr>
<tr>
<td>1195366</td>
<td>Failed to get member information.</td>
</tr>
<tr>
<td>1195367</td>
<td>Failed to sort outline.</td>
</tr>
<tr>
<td>1195377</td>
<td>IS Error: Unable to delete outline.</td>
</tr>
<tr>
<td>1195378</td>
<td>IS Error: SQL Generation for member load failed.</td>
</tr>
<tr>
<td>1195379</td>
<td>IS Error: Database close operation failed after load.</td>
</tr>
<tr>
<td>1195380</td>
<td>IS Error: Catalog close operation failed after load.</td>
</tr>
<tr>
<td>1195381</td>
<td>IS Error: Essbase close operation failed after load.</td>
</tr>
<tr>
<td>1195382</td>
<td>IS Error: No OLAP model name specified.</td>
</tr>
<tr>
<td>1195383</td>
<td>IS Error: Failed to open OLAP model.</td>
</tr>
<tr>
<td>1195384</td>
<td>IS Error: Failed to close OLAP Metadata Catalog.</td>
</tr>
<tr>
<td>1195385</td>
<td>Failed to create virtual model.</td>
</tr>
<tr>
<td>1195386</td>
<td>Member load completed with errors.</td>
</tr>
<tr>
<td>1195387</td>
<td>Member load terminated due to error.</td>
</tr>
<tr>
<td>1195389</td>
<td>Failed to get dimbuild SQL.</td>
</tr>
</tbody>
</table>

### Data Load Error Messages

Table 10 lists some of the error messages that Integration Server may generate during a data load. Italicized words represent variable names inserted in the error message. Refer to the error message number in the first column and make a note of it if you need to contact Hyperion Technical Support with a problem.

Table 10  Essbase Integration Server Data Load Error Messages

<table>
<thead>
<tr>
<th>Windows Message Number</th>
<th>Message Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>1195034</td>
<td>Cell load exceptions encountered.</td>
</tr>
<tr>
<td>1195047</td>
<td>Data load exceptions encountered. ?, amount.</td>
</tr>
<tr>
<td>1195063</td>
<td>Failed to activate Essbase application.Essbase.database.</td>
</tr>
<tr>
<td>1195082</td>
<td>Failed to execute esscmd script esscmd script.</td>
</tr>
<tr>
<td>1195369</td>
<td>Data load completed with errors.</td>
</tr>
<tr>
<td>1195370</td>
<td>Data load terminated with errors.</td>
</tr>
<tr>
<td>1195388</td>
<td>Failed to get data load SQL.</td>
</tr>
</tbody>
</table>
Drill-Through Report Error Messages

Drill-through reports are created using the Integration Services Console OLAP Metaoutline standard user interface and are viewed using Essbase Spreadsheet Add-in for Excel or Lotus 123 or any other Hyperion drill-through client.

Table 11 lists some of the error messages that Integration Server may generate to Essbase users viewing drill-through reports. Italicized words represent variable names inserted in the error message. Refer to the error message number in the first column and make a note of it if you need to contact Hyperion Technical Support with a problem.

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

Table 12 lists some of the miscellaneous error messages that Integration Server may generate during OLAP model and metaoutline creation and during data and member loads. Italicized words represent variable names inserted in the error message. Refer to the error message number in the first column and make a note of it if you need to contact Hyperion Technical Support with a problem.

<table>
<thead>
<tr>
<th>Windows Message Number</th>
<th>Message Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1195004</td>
<td><em>Metaoutline</em> failed validation. Database measures not specified.</td>
</tr>
<tr>
<td>1195007</td>
<td><em>Add job</em> failed.</td>
</tr>
<tr>
<td>1195017</td>
<td><em>Syntax</em> Syntax error at <em>character location</em> in the filter expression <em>filter</em>.</td>
</tr>
<tr>
<td>1195018</td>
<td><em>Member</em> cannot have aggregation filter. Only leaf members can have aggregation filters.</td>
</tr>
<tr>
<td>1195054</td>
<td>Error message unavailable for this error.</td>
</tr>
<tr>
<td>1195057</td>
<td>Essbase server on <em>Essbase server computer</em> is not enabled with Integration Services option. Please inform your system administrator.</td>
</tr>
<tr>
<td>1195084</td>
<td>Failed to get job info (<em>OS error Job-ID</em>).</td>
</tr>
<tr>
<td>1195086</td>
<td>Failed to get the ODBC message.</td>
</tr>
<tr>
<td>1195090</td>
<td>Failed to remove <em>temporary batch file</em> (<em>OS error operating system error</em>).</td>
</tr>
<tr>
<td>1195091</td>
<td>Failed to remove job (<em>OS error operating system error</em>).</td>
</tr>
<tr>
<td>1195092</td>
<td>Failed to remove old job (<em>OS error operating system error</em>).</td>
</tr>
<tr>
<td>1195094</td>
<td>Failed to schedule job (<em>OS error operating system error</em>).</td>
</tr>
<tr>
<td>1195096</td>
<td>Fatal error.</td>
</tr>
<tr>
<td>1195124</td>
<td>Internal system error. Please contact Technical Support with the error number 1999999.</td>
</tr>
<tr>
<td>1195127</td>
<td>Invalid command line option <em>OLAP Integration Server switch</em>.</td>
</tr>
<tr>
<td>1195128</td>
<td>Invalid datatype specification <em>data type</em>.</td>
</tr>
<tr>
<td>Windows Message Number</td>
<td>Message Contents</td>
</tr>
<tr>
<td>------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>1195133</td>
<td>Invalid session identifier; please log in to Integration Services again.</td>
</tr>
<tr>
<td>1195159</td>
<td>Replace Failed member transformation.</td>
</tr>
<tr>
<td>1195185</td>
<td>Trying to access invalid memory. Contact Technical Support.</td>
</tr>
<tr>
<td>1195187</td>
<td>Unknown ODBC exception encountered while closing database.</td>
</tr>
<tr>
<td>1195206</td>
<td>Syntax Syntax error at character location in the filter expression filter.</td>
</tr>
<tr>
<td>1195207</td>
<td>Integration Services is already running on this machine.</td>
</tr>
<tr>
<td>1195208</td>
<td>Failed to release locks.</td>
</tr>
<tr>
<td>1195333</td>
<td>ODBC Error: Encountered unknown ODBC exception while opening database.</td>
</tr>
<tr>
<td>1195334</td>
<td>ODBC Error: Encountered unknown ODBC exception while closing database.</td>
</tr>
<tr>
<td>1195336</td>
<td>ODBC Error: Failed to get ODBC Error message for error number.</td>
</tr>
<tr>
<td>1195338</td>
<td>IS Error: Error during parsing Load Parameters.</td>
</tr>
<tr>
<td>1195343</td>
<td>Essbase Error: Essbase Server on Essbase server computer is not enabled with Integration Services option.</td>
</tr>
<tr>
<td>1195344</td>
<td>Failed to unload database.</td>
</tr>
<tr>
<td>1195348</td>
<td>Failed to get information for Database database.</td>
</tr>
<tr>
<td>1195391</td>
<td>IS Error: Unable to get tables for pattern name SQL regular expression pattern from data source.</td>
</tr>
<tr>
<td>1195392</td>
<td>IS Error: Unable to get columns for table data source table from data source.</td>
</tr>
<tr>
<td>1195393</td>
<td>IS Error: Unable to extract foreign keys for tables data source table, data source table from data source.</td>
</tr>
<tr>
<td>1195394</td>
<td>IS Error: Failed to bind columns.</td>
</tr>
<tr>
<td>1195395</td>
<td>IS Error: Failed to fetch next row.</td>
</tr>
<tr>
<td>1195396</td>
<td>IS Error: Failed to get number of columns.</td>
</tr>
<tr>
<td>1195397</td>
<td>IS Error: Failed to get column attributes.</td>
</tr>
<tr>
<td>1195398</td>
<td>IS Error: Failed to get statement handle.</td>
</tr>
<tr>
<td>1195399</td>
<td>IS Error: Failed to release statement handle.</td>
</tr>
<tr>
<td>1195400</td>
<td>IS Error: Failed to get number of rows.</td>
</tr>
<tr>
<td>1195401</td>
<td>IS Error: Unable to delete the model model name.</td>
</tr>
</tbody>
</table>
Table 13 lists the most common errors that you will find in the `dataload.txt` file. The `dataload.txt` file for a specific data load is located in a subfolder in the \ISHOME\loadinfo directory in Windows and `${ISHOME}/loadinfo` in UNIX. In the `loadinfo` directory, each data load generates a subfolder named in the following format:

<application_database_timestamp_sessionnumber>

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

Table 13  Essbase Messages Generated During Loads

<table>
<thead>
<tr>
<th>Message Number</th>
<th>Message Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>3303</td>
<td>Member not found in database.</td>
</tr>
<tr>
<td>3304</td>
<td>Insufficient access to store data.</td>
</tr>
<tr>
<td>3333</td>
<td>Bad data value supplied.</td>
</tr>
<tr>
<td>3335</td>
<td>Record rejected because of duplicate member names.</td>
</tr>
<tr>
<td>3336</td>
<td>Member/Data unknown.</td>
</tr>
<tr>
<td>3337</td>
<td>Record rejected because of dimension conflicts with Header Name.</td>
</tr>
</tbody>
</table>
Integration Services Limits and Guidelines

In This Appendix

Artifact Limits .......................................................................................................................... 105
Source Database Artifact Naming Limits .................................................................................. 108
Setting the Compression Dimension ....................................................................................... 109
Hybrid Analysis Guidelines ..................................................................................................... 110
Using Advanced Relational Access ......................................................................................... 114
Drill-through Report Guidelines ............................................................................................. 117
Unicode Guidelines ................................................................................................................ 118
Duplicate Member Names Guidelines ....................................................................................... 118
DB2 Cube Views Restrictions .................................................................................................. 119
Unsupported Aggregate Functions in SAP BW ........................................................................ 120

This appendix describes Integration Services artifacts and database naming limits; Hybrid Analysis, Drill-through, and Unicode guidelines; restrictions on DB2 cube views; and aggregate functions not supported in SAP BW.

You should be familiar with these limits and guidelines before you begin using Integration Services.

For restrictions and guidelines on naming artifacts, see Chapter 6, “Naming Restrictions for Essbase Applications, Databases, and Members.” See also Integration Services Console online help.

Artifact Limits

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

Note:

The database artifact naming limits specified in Table 16 also apply to SAP BW Info Objects (QueryCubes and InfoCubes).
<table>
<thead>
<tr>
<th>Artifact</th>
<th>Limit for Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Names and Related Fields</td>
<td></td>
</tr>
<tr>
<td>Alias name</td>
<td>Non-Unicode application limit: 80 bytes&lt;br&gt;Unicode-mode application limit: 80 characters</td>
</tr>
<tr>
<td>Alias table name</td>
<td>Non-Unicode application limit: 30 bytes&lt;br&gt;Unicode-mode application limit: 30 characters</td>
</tr>
<tr>
<td>Essbase Server name</td>
<td>Non-Unicode application limit: 29 bytes&lt;br&gt;Unicode-mode application limit: 50 characters</td>
</tr>
<tr>
<td>Application name</td>
<td>Non-Unicode application limit: 8 bytes&lt;br&gt;Unicode-mode application limit: 30 characters</td>
</tr>
<tr>
<td>Application description</td>
<td>Non-Unicode application limit: 79 bytes&lt;br&gt;Unicode-mode application limit: 80 characters</td>
</tr>
<tr>
<td>Custom-defined function name</td>
<td>Non-Unicode application limit: 127 bytes. MaxL and the API truncate characters after 127 bytes.</td>
</tr>
<tr>
<td>Custom-defined macro name</td>
<td>Unicode-mode application limit: 128 characters. MaxL and the API truncate characters after 128 characters.</td>
</tr>
<tr>
<td>Custom-defined function specification</td>
<td>In either case, no truncation on server. No error is displayed if truncation occurs.</td>
</tr>
<tr>
<td>Custom-defined macro specification</td>
<td></td>
</tr>
<tr>
<td>Custom-defined function and macro comment</td>
<td>Non-Unicode application limit: 255 bytes. After 255 bytes, characters are truncated by MaxL and API.</td>
</tr>
<tr>
<td>Data source name</td>
<td>Non-Unicode application limit: 46 characters</td>
</tr>
<tr>
<td>Database name</td>
<td>Non-Unicode application limit: 8 bytes&lt;br&gt;Unicode-mode application limit: 30 characters</td>
</tr>
<tr>
<td>Database description</td>
<td>Non-Unicode application limit: 79 bytes&lt;br&gt;Unicode-mode application limit: 80 characters</td>
</tr>
<tr>
<td>Directory path</td>
<td>Non-Unicode application limit: 256 bytes&lt;br&gt;Unicode-mode application limit: 1024 bytes</td>
</tr>
<tr>
<td>For example: /AIS/bin</td>
<td></td>
</tr>
<tr>
<td>File names for calculation scripts, report scripts, and rules files</td>
<td>Non-Unicode application limit: 8 bytes&lt;br&gt;Unicode-mode application limit: If included within a path, the smaller of the following two values:</td>
</tr>
<tr>
<td></td>
<td>● 1024 bytes&lt;br&gt;● The limit established by the operating system</td>
</tr>
<tr>
<td><strong>Artifact</strong></td>
<td><strong>Limit for Applications</strong></td>
</tr>
<tr>
<td>------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Filter name</td>
<td>If not included within a path, as in some MaxL statements, 1024 bytes. Non-Unicode application limit: 30 bytes Unicode-mode application limit: 30 characters</td>
</tr>
<tr>
<td>Group name</td>
<td>Non-Unicode application limit: 30 bytes Unicode-mode application limit: 30 characters</td>
</tr>
<tr>
<td>Linked reporting artifact cell note</td>
<td>599 bytes</td>
</tr>
<tr>
<td>Linked reporting artifact URL</td>
<td>512 characters (always single-byte characters)</td>
</tr>
<tr>
<td>Member comment field</td>
<td>Non-Unicode application limit: 255 bytes Unicode-mode application limit: 256 characters</td>
</tr>
<tr>
<td>Member comment field (extended)</td>
<td>8192 bytes</td>
</tr>
<tr>
<td>Member name</td>
<td>Non-Unicode application limit: 80 bytes Unicode-mode application limit: 80 characters</td>
</tr>
<tr>
<td>OLAP model name</td>
<td>80 characters</td>
</tr>
<tr>
<td>OLAP metaoutline name</td>
<td>80 characters</td>
</tr>
<tr>
<td>Password</td>
<td>Non-Unicode application limit: 100 bytes Unicode-mode application limit: 100 characters</td>
</tr>
<tr>
<td>Substitution variable name</td>
<td>80 bytes</td>
</tr>
<tr>
<td>Substitution variable value</td>
<td>255 bytes</td>
</tr>
<tr>
<td>Trigger name</td>
<td>30 bytes</td>
</tr>
<tr>
<td>User-defined query</td>
<td>No limitations</td>
</tr>
<tr>
<td>User names</td>
<td>Non-Unicode application limit: 30 bytes Unicode-mode application limit: 30 characters</td>
</tr>
<tr>
<td>Variable names</td>
<td>32 bytes</td>
</tr>
<tr>
<td>Data Load and Dimension Building Limits</td>
<td></td>
</tr>
<tr>
<td>Data load query</td>
<td>No limitations</td>
</tr>
<tr>
<td>Member load query</td>
<td>No limitations</td>
</tr>
<tr>
<td>Number of alias tables associated with a metaoutline</td>
<td>9, including the Default alias table</td>
</tr>
<tr>
<td>Number of error messages written to a data load or dimension build error log (DATAERRORLIMIT in essbase.cfg)</td>
<td>Default 1000, minimum 1, maximum 65000</td>
</tr>
</tbody>
</table>
### Limit for Applications

<table>
<thead>
<tr>
<th>Artifact</th>
<th>Limit for Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selection and rejection criteria</td>
<td>Number of characters that describe selection and rejection criteria: combination of all criteria limited to 32 KB</td>
</tr>
</tbody>
</table>

### Other Limits

<table>
<thead>
<tr>
<th>Caches: data, data file, index</th>
<th>2 GB</th>
</tr>
</thead>
</table>
| Formula size                   | ● Created in Formula Editor: 64 KB. Formulas in calculation scripts are not subject to this limit.  
                                  | ● Created in MaxL, using multi-byte characters: 40 KB. Formulas in calculation scripts are not subject to these limits. |
| Number of security filters     | Per Essbase Server, 65535                                     |
|                                | Per Essbase database, 32290                                   |
| Number of users                | 30,000. Errors can occur if you create more than 30,000 users. |
| Number of members in an Essbase outline | Approximately 1,000,000 explicitly defined in an Essbase outline for block storage  
                                          | Approximately 20,000,000 explicitly defined in an Essbase outline for aggregate storage  
                                          | Hybrid Analysis and some uses of partitions enable access to many more members than are explicitly listed in an outline, the actual number of members accessible through the database is much higher.  
                                          | Longer names, which often occur if multi-byte characters are used, decrease the number of members that are allowed. |

### Source Database Artifact Naming Limits

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

**Note:**

Table names may contain blank spaces. Column names may contain blank spaces and question marks.

**Note:**

Table and column names may be greater than 30 characters in length.
### Setting the Compression Dimension

In aggregate storage databases, the size of the compressed database changes depending on which dimension is specified as the compression dimension, and can affect retrieval performance.

When the data model property is set to aggregate storage, the Accounts dimension is specified as the compression dimension by default. However, you may determine that another dimension is the optimal choice for compression. Using the Compression Dimension check box, you can specify any single dimension as the compression dimension.

After performing a data load, use Administration Services Console or MaxL to view detailed compression and query statistics for your database.

### Guidelines

- The compression dimension option is ignored if the outline is being built with Essbase versions earlier than 9.3. In versions less than 9.3, the Accounts dimension is tagged as the compression dimension internally.
- The following dimensions cannot be specified as the compression dimension:
  - Multiple hierarchy dimensions
  - Attribute dimensions
  - A base dimension with an attribute association
● If you tag a compression dimension with an Outline Hierarchy Information option other than “Dynamic at Dimension Level,” that dimension will be tagged as a dynamic dimension when it is loaded into Essbase.

● The compressions dimension option applies to metaoutlines specified for aggregate storage only. To facilitate switching between aggregate and block data storage models, the option is always available, whether the metaoutline is specified for block storage or aggregate storage. If the metaoutline is specified for block storage, this option is ignored.

### Hybrid Analysis Guidelines

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

#### Data Source Guidelines

● A single Essbase database can be associated with only one hybrid analysis relational data source.

● A hybrid analysis data source can consist of only one relational database.

● Hybrid Analysis does not support SAP BW data warehouses.

● Hybrid Analysis supports data that is stored using either block storage or aggregate storage.

● Hybrid Analysis supports Unicode-enabled data sources.

#### Dimensions Guidelines

● Hybrid Analysis is not supported on accounts dimensions.

● If the time dimension contains hybrid analysis-enabled members, the time dimension does not support Dynamic Time Series.

● Hybrid Analysis is not supported on user-defined dimensions.

● In an outline that is hybrid analysis-enabled, you can perform operations and analyses on dimensions that have attributes attached to one or more levels. The attribute dimension should be fully loaded into Essbase.

● Only the first hierarchy of a dimension with alternate hierarchies can have members enabled for hybrid analysis on its lowest levels.

● When building a dimension that is enabled for hybrid analysis, you must ensure that the column in the data source table that contributes to the leaf level of the Essbase portion of the dimension is non-nullable.

#### Members Guidelines

● Only the lowest level members of a dimension can be enabled for hybrid analysis.
You should not rename a hybrid analysis-enabled member. If you rename a member, the member may not be retrieved the next time you perform a drill-through operation.

Hybrid Analysis supports only parent-child prefixing on member names.

Essbase does not support aliases for members that are enabled for hybrid analysis.

Hybrid Analysis does not support scaling of measures dimension members using any of the operators + (addition), - (subtraction), * (multiplication), and / (division). If you use the scaling operators, drill-through queries into hybrid analysis data may show a mismatch between aggregated level-0 values in the Essbase database and the corresponding detail values in your data source.

Essbase ignores all member properties, such as formulas, UDAs, and aliases for members that are enabled for hybrid analysis.

You can associate an attribute member with a member enabled for hybrid analysis but you must make the association by metaoutline member name and not by level.

Essbase supports drill-through operations defined on members that are enabled for Hybrid Analysis.

You cannot apply properties of the Account Info tab to Hybrid Analysis members. Specifically, Time Balance, Skip, Variance Reporting, or Currency Conversion properties are not supported for Hybrid Analysis-enabled members.

**Spreadsheet Guidelines**

Hybrid Analysis does not return numeric data in a spreadsheet if the member from the Accounts dimension is part of a ragged hierarchy.

Hybrid Analysis is not supported with the Member Selection feature. You cannot select Hybrid Analysis members from the Member Selection dialog box.

If you have multiple levels of hybrid analysis members in your outline, performing a zoom out operation on the bottom level hybrid analysis member takes you directly to the Essbase parent member, bypassing all other hybrid analysis levels.

**Operations Guidelines**

Hybrid Analysis supports Dynamic Time Series.

Essbase requires the OLAP Metadata Catalog created in Integration Services in order to drill down in a hybrid analysis data source.

Hybrid Analysis does not support transparent, replicated, or linked partitions.

Hybrid Analysis supports recursive hierarchies.

**Limitations on Transformation Rules with Hybrid Analysis**

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

Transformations Not Supported by Hybrid Analysis

Hybrid Analysis does not support the following transformations:
● Dropping spaces from around a member name
● Applying a prefix without a separator
● Applying names prefixes for all parents with or without a separator
● Applying a suffix without a separator
● Applying all suffixes of parent names with or without a separator
● Applying scaling to measures

Limitations on Using Formulas with Hybrid Analysis

Formulas used with hybrid analysis-enabled members are subject to the following limitations:
● Formulas are supported only on a measures dimension.
● Formulas cannot be attached to relational members.
● Formulas cannot reference a relational member by name.
● Member set functions (such as @CHILDREN and @DESCENDANTS), which generate member lists in a formula, execute only in the Essbase portion of the outline.

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

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

Unsupported Essbase Functions in Hybrid Analysis

Hybrid Analysis does not support all Essbase functions. The following topics specify the categories of significant Essbase functions not supported by Hybrid Analysis.
**Relationship Functions**

Hybrid Analysis does not support functions that look up specific values in the database based on current cell location and a series of parameters. Examples:

<table>
<thead>
<tr>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>@ANCEST</td>
</tr>
<tr>
<td>@SPARENT</td>
</tr>
<tr>
<td>@SANCEST</td>
</tr>
<tr>
<td>@CURLEV</td>
</tr>
<tr>
<td>@PARENT</td>
</tr>
<tr>
<td>@CURGEN</td>
</tr>
</tbody>
</table>

**Member Condition Functions That Use Boolean Test**

Hybrid Analysis does not support functions used to specify member conditions. Examples:

<table>
<thead>
<tr>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>@ISIANCEST</td>
</tr>
<tr>
<td>@ISLEV</td>
</tr>
<tr>
<td>@ISIPARENT</td>
</tr>
<tr>
<td>@ISSAMEGEN</td>
</tr>
<tr>
<td>@ISISIBLING</td>
</tr>
<tr>
<td>@ISUDA</td>
</tr>
</tbody>
</table>

**Range Functions**

Hybrid Analysis does not support functions that take a range of members as arguments. Rather than return a single value, these functions calculate a series of values internally based on the range specified. Examples:

<table>
<thead>
<tr>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>@PRIOR</td>
</tr>
<tr>
<td>@MOVAVG</td>
</tr>
<tr>
<td>@SHIFT</td>
</tr>
<tr>
<td>@ALLOCATE</td>
</tr>
<tr>
<td>@PRIORS</td>
</tr>
<tr>
<td>@MDALLOCATE</td>
</tr>
<tr>
<td>@SHIFTS</td>
</tr>
<tr>
<td>@VAR</td>
</tr>
<tr>
<td>@NEXT</td>
</tr>
<tr>
<td>@VARPER</td>
</tr>
<tr>
<td>@MDSHIFT</td>
</tr>
<tr>
<td>@MEDIAN</td>
</tr>
<tr>
<td>@MOVSUM</td>
</tr>
<tr>
<td>@RANK</td>
</tr>
</tbody>
</table>

**Attribute Functions**

Hybrid Analysis does not support any Essbase functions that deal with attributes. Examples:

<table>
<thead>
<tr>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>@ATTRIBUTEVAL</td>
</tr>
<tr>
<td>@WITHATTR</td>
</tr>
<tr>
<td>@ATTRIBUTESVAL</td>
</tr>
</tbody>
</table>
Current Member and XREF
Hybrid Analysis does not support the following functions used to determine whether the current member is the member being specified:

@CURRMBR  @XREF

Using Advanced Relational Access
By default, when Integration Server creates an Essbase outline, it loads, or builds, all member levels specified in the metaoutline into a multidimensional database. You can, however, set Integration Server to build to a specified member level (Hybrid Analysis) or build only to the dimension level (Advanced Relational Access). Building down to a specified level produces a smaller Essbase outline and a smaller multidimensional database. Smaller databases can be useful for users with limited disk space who do not need to see the lowest level of detail.

Integration Services uses Advanced Relational Access to give Essbase end users direct access to data from relational databases or data warehouses. This feature enables users to perform online analytical processing (OLAP) on very large data sets.

In Integration Services Console, Advanced Relational Storage is enabled at the metaoutline level. When the Relational Storage option is selected, all members of all non-accounts dimensions are automatically enabled for relational storage. Alternatively, you can enable relational storage on selected non-accounts dimensions.

When a metaoutline is enabled for Advanced Relational Access, end users are able to query directly on relationally-stored members. Essbase Server issues SQL queries to retrieve data from the database or data warehouse. All members of the dimension are accessed directly from the relational data source.

Note:
For detailed information on enabling Advanced Relational Access, see Integration Services Console online help.

Advanced Relational Access Guidelines
When users construct Advanced Relational Access queries, the points outlined in the following sections should be noted.

General Guidelines
Here are general guidelines to keep in mind when using Advanced Relational Access:

- Only outlines that are valid for aggregate storage can be Advanced Relational Access outlines. Block storage is not supported.
- Dynamic Time Series members are not supported.
Advanced Relational Access requires MDX formulas.

**Note:**
An MDX query made against a ragged hierarchy in an Essbase cube returns data results that can differ significantly from the results obtained when the same MDX query is made against a ragged hierarchy in an Advanced Relational Access cube. See the section “Different Values Loaded in Ragged Hierarchies” on page 117.

- Members enabled for Advanced Relational Access are shown in Administration Services Console Outline Viewer but are not shown in the Outline Editor.
- A Time dimension from the Fact table is not supported. Hyperion strongly recommends that you create a separate Time dimension table with time data only.
- When there are multiple measures defined in a metaoutline, consolidation at the highest level is based on the first measure only.

**Data Source Guidelines**
Here are guidelines to keep in mind when considering your data source for Advanced Relational Access:

- Advanced Relational Access cannot be enabled in metaoutlines that are connected to multiple data sources.
- Star and snowflake schemas are supported.
- Teradata RDBMS provides different types of SQL date types, but Advanced Relational Access only supports the ‘DATE’ SQL date type.

**Dimension Guidelines**
Here are guidelines to keep in mind when working with dimensions in Advanced Relational Access:

- Advanced Relational Access does not support recursive dimensions/hierarchies.
- Alternate hierarchies in standard dimensions are not allowed.
- Attribute dimensions are ignored.
- User-defined dimensions are not supported.
- When a dimension is enabled for relational storage, Integration Server builds only to the dimension level. All members of the dimension are accessed directly from the relational data source.
- Accounts dimension must be created from the fact table.
- For the accounts dimension, no hierarchy is supported. All members should be of generation 2.
- Aliases are not supported.
**Member Guidelines**

Here are guidelines to keep in mind when working with members in Advanced Relational Access:

- The first member of the accounts dimension must be a base or stored measure. It cannot be a user-defined member.
- Aliases are not supported.
- Duplicate members are not supported.
- Shared members are not supported.
- Relational members are not automatically expanded when part of an outline is expanded. To view the relational members, you must manually click the member node containing the relational members.
- Relational members are not automatically collapsed when part of an outline is collapsed. To collapse the relational members, you must manually click the member node containing the relational members.
- In Integration Services Console, you can specify multiple sort order columns for each generation, selecting ascending or descending order.
- In Integration Services Console, you can associate a key column with each generation name and tag the key column as unique.

**Unsupported Data Types**

The following data types are not supported in Advanced Relational Access:

- DB2: REAL
- Oracle: NCHAR and NVARCHAR2
- SQL Server: TINYINT and SMALLINT

**Note:**

The data types TINYINT and SMALLINT cannot be used as a measure; however, they can be used as members.

- Teradata: FLOAT

**Note:**

Your member set column names should not be based on columns of FLOAT data types.

**Essbase Databases and Advanced Relational Access**

When you use Advanced Relational Access, your Essbase database has the following characteristics:

- The accounts dimension completely resides in Essbase.
For all non-accounts dimensions, only the root members (the dimension level) reside in the Essbase database. All other non-accounts members are accessed directly from the relational database.

**Different Values Loaded in Ragged Hierarchies**

There is a significant difference in the way Advanced Relational Access and Essbase perform data loads in ragged hierarchies. Because of this difference, separate queries made against a ragged hierarchy can display different consolidation totals.

Essbase loads values in the upper-level members of a ragged hierarchy only when the level 0 members are non-null. Advanced Relational Access, however, loads values in the upper level members of the ragged hierarchy regardless of the value in the level 0 member.

**Drill-through Report Guidelines**

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

- Drill-through operations can be performed on a Unicode Essbase database.
- Drill-through reports may be directed to an alternate Integration Server. Select the alternate server in the OLAP Metaoutlne Properties dialog box in Integration Services Console.
- Drill-through operations can be performed with Dynamic Time Series.
- Drill-through operations can be performed on an alternate data source. An alternate data source is a source other than the primary or secondary data source used to create a metaoutline. The alternate data source must contain the same data structure, including column names and data types, as the primary or secondary data source originally used to create the report. Drill-through operations can be performed on an alternate data source that is in a second language.
- If you create drill-through reports for a metaoutline based on multiple data sources, the SQL used for each drill-through report cannot include a join across a data source boundary; that is, each drill-through report can be based on only one data source. A metaoutline can contain multiple drill-through reports but all must be based on a single data source.
- Multi-cell drill-through is supported under the following conditions:
  - All members selected for multi-cell drill-through come from the same physical table and column in the database.
  - All members selected for multi-cell drill-through come from the same level in the metaoutline.
  - The selected members cannot come from more than one hierarchy.
- The metaoutline in which you are designing a drill-through report cannot contain alternate recursive hierarchies. For example, you cannot insert the same member from a recursive hierarchy into a dimension more than once.
• You cannot use the $$ substitution variable in the template SQL (user-defined SQL) for a drill-through report when the intersection level of the dimension is defined at Generation 1 and the dimension is built from a parent/child table.
• Drill-through operations can be performed on an database containing multibyte character sets.

Unicode Guidelines

• You must use the manual catalog creation procedures to create a Unicode-enabled OLAP Metadata Catalog, as described in Essbase Integration Services Installation Guide.
• Drill-through operations are not supported on a Unicode Essbase database.
• Integration Services does not support Unicode databases in SQL Server, Sybase and Informix RDBMSs.
• If the database is UTF-8 but you are using only one language, select the UTF-8 option from the Code Page drop-down list whenever you access the database with the Login, OLAP Metadata Catalog Setup, Set Login Defaults, and Data Source dialog boxes.
• Unicode is supported in SAP BW data sources. If the data in your SAP BW data source is not in English, select the UTF-8 option from the Code Page drop-down list whenever you access the database.

Duplicate Member Names Guidelines

When a metaoutline is loaded into Essbase Server, an Essbase outline is usually created with each member name being unique. This is the default behavior of Integration Server. You can change this default and specify a metaoutline that, when loaded into an Essbase database, creates an Essbase outline containing duplicate (non-unique) member names, as shown in the following example:

US-->Massachusetts-->Springfield
US-->Missouri-->Springfield

Integration Services generates a unique internal identifier, which is applied to any duplicate member name in a metaoutline. This process enables support for duplicate member names during member loads, data loads, and drill-through operations.

When using duplicate member names, keep in mind the guidelines described in the sections that follow:

Dimensions in Duplicate Outlines

• Dimension names must be unique.
• In a dimension, a user-defined member cannot have the same name as a measure.
Members in Duplicate Outlines

- Duplicate member names are not allowed under the same parent. This is also true for user-defined members.
- A member cannot have children with duplicate names.
- Duplicate members may be shared or non-shared members.
- Duplicate members are supported in both aggregate storage and block storage.
- Duplicate member names are not supported if there are multiple data sources.
- Duplicate member names support duplicate aliases.

Other Guidelines for Duplicate Outlines

- Drill-through operations containing duplicate member names are supported.
- Exercise caution if you choose to override database SQL commands. You may also have records rejected in error. It is especially recommended that you not change column names in SQL.
- Duplicate outlines support MDX commands.

Note:

For detailed information on creating an Oracle’s Hyperion® Essbase® – System 9 outline with duplicate member names, see Integration Services Console online help.

DB2 Cube Views Restrictions

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

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

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

  \[ \text{physicalTable} = \text{"CubeView(NoTable)"} \]
Elements with a Cube View prefix need to be mapped and defined by the Integration Services/CubeView administrator as appropriate.

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

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

- A CubeView XML must contain at least one CubeModel.
- A CubeView XML cannot contain multiple cubes. It may contain one cube or zero cubes.

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

### Unsupported Aggregate Functions in SAP BW

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

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

The aggregate function is defined at the measures dimension. If a measures dimension contains an aggregate function not supported in Oracle's Essbase® Integration Services, the aggregate function is mapped to <NONE>.
**accounts dimension** A dimension type that makes accounting intelligence available. Only one dimension can be defined as Accounts.

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

**Advanced Relational Access** The integration of a relational database with an Essbase multidimensional database so that all data remains in the relational database and is mapped to summary-level data residing in the Essbase database.

**aggregate cell** A cell comprising several cells. For example, a data cell that uses Children(Year) expands to four cells containing Quarter 1, Quarter 2, Quarter 3, and Quarter 4 data.

**aggregate storage database** The database storage model designed to support large-scale, sparsely distributed data which is categorized into many, potentially large dimensions. Upper level members and formulas are dynamically calculated, and selected data values are aggregated and stored, typically with improvements in overall aggregation time.

**aggregate view**

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

**aggregation level** See *consolidation level*.

**alias** An alternative name. For example, for a more easily identifiable column descriptor you can display the alias instead of the member name.

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

**alias table** A table that contains alternate names for members.

**alternate hierarchy** A hierarchy of shared members. An alternate hierarchy is based upon an existing hierarchy in a database outline, but has alternate levels in the dimension. An alternate hierarchy allows the same data to be seen from different points of view.

**alternate name** See *alias*.

**ancestor** A branch member that has members below it. For example, the members Qtr2 and 2006 are ancestors of the member April.

**application** (1) A software program designed to run a specific task or group of tasks such as a spreadsheet program or database management system. (2) A related set of dimensions and dimension members that are used to meet a specific set of analytical and/or reporting requirements.

**Architect** See *Essbase Integration Services Console*.

**attribute** Characteristics of a dimension member. For example, Employee dimension members may have attributes of Name, Age, or Address. Product dimension members can have several attributes, such as a size and flavor.

**base dimension** A standard dimension that is associated with one or more attribute dimensions. For example, assuming products have flavors, the Product dimension is the base dimension for the Flavors attribute dimension.

**block** The primary storage unit which is a multidimensional array representing the cells of all dense dimensions.
block storage database  The Essbase database storage model categorizing and storing data based on the sparsity of data values defined in sparse dimensions. Data values are stored in blocks, which exist only for sparse dimension members for which there are values.

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

Builder  See Essbase Integration Services Console.

catalog  See OLAP Metadata Catalog.

cell  (1) The data value at the intersection of dimensions in a multidimensional database; the intersection of a row and a column in a worksheet. (2) A logical group of nodes belonging to one administrative domain.

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

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

column  A vertical display of information in a grid or table. A column can contain data from one field, derived data from a calculation, or textual information.

Command Interface  See Essbase Integration Services Shell.

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

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

consolidation  The process of aggregating data from dependent entities to parent entities. For example, if the dimension Year consists of the members Qtr1, Qtr2, Qtr3, and Qtr4, its consolidation is Year.

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

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

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

data file  A file containing data; Essbase generates data files during data loads and adds additional data files during aggregations.

data load rules  A set of criteria that determines how to load data from a text-based file, a spreadsheet, or a relational data set into a database.

data value  See cell.

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.

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  In block storage databases, a dimension likely to contain data for every combination of dimension members. For example, time dimensions are often dense because they can contain all combinations of all members. Contrast with sparse dimension.

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

detail member  See normalization.
**dimension**  A data category used to organize business data for retrieval and preservation of values. Dimensions usually contain hierarchies of related members grouped within them. For example, a Year dimension often includes members for each time period, such as quarters and months.

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

**dimension build rules**  In Essbase, 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**  (1) A table that includes numerous attributes about a specific business process. (2) In Essbase Integration Services, a container in the OLAP model for one or more relational tables that define a potential dimension in Essbase.

**dimension type**  A dimension property that enables the use of predefined functionality. Dimensions tagged as time have a predefined calendar functionality.

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

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

**duplicate member name**  The multiple occurrence of a member name in a database, with each occurrence representing a different member. For example, a database has two members named “New York.” One member represents New York state and the other member represents New York city.

**duplicate member outline**  A database outline containing duplicate member names.

**Dynamic Calc and Store members**  A member in a block storage outline that Essbase calculates only upon the first retrieval of the value. Essbase then stores the calculated value in the database. Subsequent retrievals do not require calculating.

**Dynamic Calc members**  A member in a block storage outline that Essbase calculates only at retrieval time. Essbase discards calculated values after completing the retrieval request.

**dynamic calculation**  In Essbase, a calculation that occurs only when you retrieve data on a member that is tagged as Dynamic Calc or Dynamic Calc and Store. The member’s values are calculated at retrieval time instead of being precalculated during batch calculation.

**Dynamic Time Series**  A process that performs period-to-date reporting in block storage databases.

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

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

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

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

**Essbase OLAP Server**  See locale.

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

essbase.cfg  An optional configuration file for Essbase. Administrators may edit this file to customize Essbase Server functionality. Some configuration settings may also be used with Essbase clients to override Essbase Server settings.

ESSLANG  The Essbase environment variable that defines the encoding used to interpret text characters. See also encoding.

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

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

file delimiter  Characters, such as commas or tabs, that separate fields in a data source.

filter  A constraint on data sets that restricts values to specific criteria; for example, to exclude certain tables, metadata, or values, or to control access.

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

formula  In Essbase, a combination of operators and functions as well as dimension names, member names, and numeric constants. Formulas are used to calculate relationships between members of an 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 database. Generations are ordered incrementally from the top member of the dimension (generation 1) down to the child members.

generation name  A unique name that describes a generation.

Hybrid Analysis  An analysis mapping low-level data stored in a relational database to summary-level data stored in Essbase, combining the mass scalability of relational systems with multidimensional data.

Hyperion Integration Server Desktop  See Essbase Integration Services Console.

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

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

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

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

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

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

Intelligent Help  In Essbase Integration Services, procedural help displayed in a dockable window that accompanies the OLAP Model and OLAP Metaoutline main windows. Intelligent Help provides numbered procedures and links to new automatic-detection options and frequently used functions.
**intersection level** In Essbase Spreadsheet Add-in, an 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** A link between two relational database tables or topics based on common content in a column or row. A join typically occurs between identical or similar items within different tables or topics. For example, a record in the Customer table is joined to a record in the Orders table because the Customer ID value is the same in each table.

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

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

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

**level** A layer in a hierarchical tree structure that defines database member relationships. Levels are ordered from the bottom dimension member (level 0) up to the parent members.

**level 0 block** A data block for combinations of sparse, level 0 members.

**level 0 member** A member that has no children.

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

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

**locale** A computer setting that specifies a location’s language, currency and date formatting, data sort order, and the character set encoding used on the computer. Essbase uses only the encoding portion. See also encoding and ESSLANG.

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

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

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

**MDDB** See multidimensional database.

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

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

**member** A discrete component within a dimension. A member identifies and differentiates the organization of similar units. For example, a time dimension might include such members as Jan, Feb, and Qtr1.

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

**member load** In Essbase Integration Services, the process of adding dimensions and members (without data) to Essbase outlines.

**metadata** A set of data that defines and describes the properties and attributes of the data stored in a database or used by an application. Examples of metadata are dimension names, member names, properties, time periods, and security.

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

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

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

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

**multidimensional database** A method of organizing, storing, and referencing data through three or more dimensions. An individual value is the intersection point for a set of dimensions.

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

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

**ODBC** Open Database Connectivity. A database access method used from any application regardless of how the database management system (DBMS) processes the information.

**OLAP Architect** See *Essbase Integration Services Console.*

**OLAP Builder** See *Essbase Integration Services Console.*

**OLAP Catalog** See *OLAP Metadata Catalog.*

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

**OLAP Integration Server** See *Essbase Integration Server.*

**OLAP Metadata Catalog** In Essbase Integration Services, a relational database containing metadata describing the nature, source, location, and type of data that is pulled from the relational data source.

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

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

**online analytical processing (OLAP)** A multidimensional, multiuser, client-server computing environment for users who analyze consolidated enterprise data in real time. OLAP systems feature drill-down, data pivoting, complex calculations, trend analysis, 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 database structure of a multidimensional database, including all dimensions, members, tags, types, consolidations, and mathematical relationships. Data is stored in the database according to the structure defined in the outline.
pass-through transformations In Essbase Integration Services, a feature that allows you to use functions specific to your relational database management system (RDBMS) to extract data values for columns. You can provide a statement that Essbase Integration Server passes through to your RDBMS as a part of the SQL SELECT statement. You provide the statement as a property of a column and the pass-through feature returns a value for the column.

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

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

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

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

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

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

query governor An Essbase Integration Server parameter or Essbase Server configuration setting that controls the duration and size of queries made to data sources.

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

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

relational database A type of database that stores data in related two-dimensional tables. Contrast with multidimensional database.

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

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

roll-up See consolidation.

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

shared member A member that shares storage space with another member of the same name, preventing duplicate calculation of members that occur multiple times in an Essbase outline.

sibling A child member at the same generation as another child member and having the same immediate parent. For example, the members Florida and New York are children of East and each other’s siblings.

sparse dimension In block storage databases, a dimension unlikely to contain data for all member combinations when compared to other dimensions. For example, not all customers have data for all products.

SQL See Structured Query Language (SQL).

SQL Override In Essbase Integration Services, a function that enables editing of the standard SQL statements generated by Integration Server for drill-through reports. ODBC SQL, Native SQL, and stored procedures can be used when editing the standard SQL. The edited, user-defined SQL can be selected to improve data load performance when loading data into an Essbase database.

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

standard dimension A dimension that is not an attribute dimension.

star schema A logical model that represents your relational data in a form that mirrors that of OLAP data. A star schema contains a fact table and one or more dimension tables.
string A sequence of characters treated as a unit.

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

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

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

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

two-pass An 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 outline.

unary operator A mathematical indicator (+, -, *, /, %) associated with an outline member. The unary operator defines how the member is calculated during a database roll-up.

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

union An SQL command that is a type of join that combines the results of two SELECT statements. A union is often used to merge lists of values contained in two tables.

user-defined attribute (UDA) User-defined attribute, associated with members of an outline to describe a characteristic of the members. Users can use UDAs to return lists of members that have the specified UDA associated with them.

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

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

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

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

view In relational databases, logical table created by combining columns from one or more tables. A view can contain metadata and formatting information to query an OLAP data source.

virtual tables See user-defined tables.

virtual views See user-defined tables.

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

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

A
access
codes, for metaoutlines, 37
codes, for OLAP models, 36
exclusive access mode, 32
multi-user, 32
privileges, common problems, 40
problems, correcting, 54
setting, 35
standard access mode, 32
accessing
data sources, 39
OLAP Metadata Catalog, 39, 41
servers and databases, troubleshooting, 39
accounts
problems in the RDBMS, 44
user, administrative, 44
user, problems, 55
administration tools, Unicode-enabled, 29
administrative accounts, using, 44
aggregate functions in SAP BW, 120
aggregate storage, 110
Anaytic Services, testing connection from Integration Server to Essbase Server, 52
application, non-Unicode mode, 28
applications, Unicode-mode, defined, 27
artifact limits for source databases, 108
artifact limits in Integration Services, 105
AT service, 88

B
backtracking connection problems, 42
backups, OLAP Metadata Catalog, 63
bat extension files, 73
batch files
Integration Services Shell and, 70
running, 74
running Integration Services Shell scripts with, 73
scheduling manually, 88
batch mode, Integration Services Shell, 71
binding, requirement for DB2 databases, 57
block storage, 110
brackets, in Integration Services Shell syntax, 72

C
CB_CONCURRENCY table
column descriptions, 33
defined, 68
CB_OBJECT_TYPE column, data defined, 33
CB_VERSION table, 68
cbs extension files, 73
checking
common connection problems, 39
release number of Integration Services, 50
script syntax for loads, 73
client programs, custom, 28
client software
port number, 59
problems, correcting, 60
RDBMS, common problems, 40
RDBMS, correcting problems, 56
troubleshooting, 39
code page, defined, 27
codes, access
for metaoutlines, 37
for OLAP models, 36
command line
Integration Services Shell, 70
loading data from, 88
loading members from, 88
command syntax, Integration Services Shell, 72
command-line interface, using, 69
commands, Integration Services Shell, 75
commands, status of, 76
committing records during data load, 11
common problems
  areas of failure, 41
database-specific, 41
  with connections, 39
communication problems, 42
compatibility
  between RDBMS and RDBMS client, 57
  ODBC drivers, 58
  problems with product versions, 50
components, software, common problems, 40, 41
compression dimension, 109
concurrent users, 33
configuration problems (ODBC), 42
connecting
  database client to RDBMS, 44
  problems with data source, 40, 41
  problems with OLAP Metadata Catalog, 40, 41
corrections
  areas of failure, 41
  common problems, 39
  isolating problems, 42
LOADMEMBER command, 49
OLAP Metadata Catalog, 40, 41
path through components, 43
problems, correcting, 54
problems, diagnosing, 41
problems, isolating, 42
testing, 45, 51
troubleshooting, 39
correcting
  connection problems, 39, 54
data source problems, 58
database client problems, 56
  Integration Server problems, 59
  Integration Services Console problems, 60
ODBC problems, 39, 58
password problems, 55
RDBMS problems, 56
user name problems, 55
critical server error messages, 15
cron scheduling daemon, 88
cube views, 105, 119

data loads
  checking script syntax, 73
displaying primary keys in OLAP models, 23
DISTINCT clause in drill-through SQL, 12
DNS, problems, 42
domain name server, problems, 42
drill-through reports
  error messages, 100
  guidelines, 117
  limits, 117
  tables relating to, 67
driver file, ODBC, specifying on UNIX, 49
drivers, ODBC, problems, 58
DTR_ATTRIBUTES table, 67
DTR_INFO table, 67

eis.cfg file, 8, 9
  described, 20
  storing startup switch settings, 21
encoding, defined, 27
environment variables
  common problems, 40
  for database client, problems, 44, 57
  ODBC problems, 45
error messages
  data load, 99
  drill-through reports, 100
  member load, 96
  miscellaneous server messages, 101
  setting logging level, 15
Essbase
  Essbase Server, connecting to, 89
  message file, 25
  server name problems, 54
  server, message database, 15
  unsupported functions in Hybrid Analysis, 112
  users, defined, 31
Essbase message.txt file, 25
Essbase Server, connecting to, 88
Essbase Server, target data server, specifying, 78
Essbase Services
  relationship to Integration Server, 7
essbase.mdb file, 16
exclusive access mode, 32
executable name
  for Integration Server, 8
  for Integration Services Shell, 70
EXIT command, 88

extra locks
  deleting, 34
  viewing, 33

F
  file name, Essbase messages file, 13
  file names, 15
  fixing. See troubleshooting

G
  guidelines for drill-through reports, 117
  guidelines for Hybrid Analysis, 110

H
  host names, problems, 54
  Hybrid Analysis, 67
  limits on transformation rules, 111
  transformations not supported, 112
  unsupported Essbase functions, 112
  Hybrid Analysis, defined, 67
  Hybrid Analysis, tables relating to, 67
  Hyperion Connect for SAP BW, 29

I
  ignore parent during drill-through, 10
  incompatibility, problems with product versions, 50
incremental loads
  LOADALL command, 83
  LOADDATA command, 81
  LOADMEMBER command, 78
  script syntax, 75
informational messages, 15
Integration Server
  common problems, 39
  defined, 7
  determining user access, 34
  error message file, 25
  error messages, 95
  executable name, 8
  executable name in UNIX, 8
  log files, 25
  logging on to, 75, 88, 89
  logging out of, 87
  message file, 25
OLAP Metadata Catalog, information in, 63
olapisvr file, 8
port number, 17, 59, 70
prerequisites for, 7
problems, correcting, 59
reconnecting to, 24
server problems, 54
shutting down, 24, 87
starting, 7, 8, 24
starting as a Windows service, 9
starting from the command line, 8
starting from the Windows Desktop, 9
startup switches, 10
startup.bat file, 8
stopping, 24
stopping as a Windows Service, 24
stopping from command line, 24
stopping from Windows Desktop, 24
switches, 10
testing connection to Essbase Server, 52
testing connection to RDBMS, 49
users, defined, 31
Integration Server users
deleting, 31
disconnecting, 31
multiple, 33
viewing locks, 33
Integration Services
artifact limits, 105
connection path, diagram, 43
release number, checking, 50
Integration Services Console
common problems, 39
port number, 17, 59, 70
problems, correcting, 60
testing connection to Essbase Server, 53
testing connection to RDBMS, 51
Integration Services Shell
asynchronous commands, 76
batch mode, 71
command syntax, 72
commands, list, 75
description, 70
executable name, 70
executing commands, 72
interactive mode, 71
loading data with, 88
loading members with, 88
running scripts with, 73
starting, 69
using, 69
using to test connections, 49
Integration Services Shell commands
EXIT, 88
LOADALL, 83
LOADDATA, 81
LOADMEMBER, 78
LOGIN, 75
LOGOUT, 87
SETCATALOG, 78
SETSOURCE, 77
SETTARGET, 78
SHUTDOWN, 87
STATUS, 76
syntax, 72
VERSION, 76
Integration Services users
creating, 31
viewing, 31
interactive mode, Integration Services Shell, 71
is.csh file, using, 48
is.sh file, using, 48
isolating, connection problems, overview, 42
ivtestlib program, 45
JOIN_HINTS table, 65
configuration parameters, display primary keys, 23
levels of access permissions, 36
library path, common problems, 40
limits
drill-through report guidelines, 117
Hybrid Analysis guidelines, 110
Integration Services artifacts, 105
source database artifacts, 108
transformation rules with Hybrid Analysis, 111
transformations not supported by Hybrid Analysis, 112
unsupported Essbase functions in Hybrid Analysis, 112
listeners, setting, 19
LOAD commands, 71, 72
LOADALL command, described, 72, 83
LOADDATA command
  described, 72, 81
  example of, 89
loading
  data, time required, 72
  from the command line, 88
Integration Services Shell and, 88
OLAP Metadata Catalog and, 7, 63
test, members into Essbase database, 52
LOADMEMBER command
  connections made by, 49
  described, 72, 78
  example of, 88
  using to test connections, 49
locale, defined, 27
locks
  deleting, 33, 34
  exclusive access mode, 32
  metaoutlines, 32
  OLAP models, 32
  read, deleting, 34
  read, viewing, 34
  standard access mode, 32
  storage location in the OLAP Metadata Catalog, 33
  timestamps and, 33
  unreleased, 33
  viewing, 33
  working with, 31, 32
  write, deleting, 34
  write, viewing, 34
log files
  changing name and location, 13
  contents, 25
  Integration Server, viewing, 25
  messages recording level, 15
logging level, 15
LOGIN command
  described, 75
  example of, 88, 89
login problems, correcting, 55
login script
diagnosing problems, 47
problems, 44
LOGOUT command, 87

M
managing
  locks and permissions, 31
  user names and passwords, 55
MB_FILTER table, 64
MB_TX_RULE table, 64
member loads
  checking script syntax, 73
  command line, 88
  error messages, 96
  LOADMEMBER command, 78
  OLAP Metadata Catalog and, 7, 63
  scripts, recording, 73
Merant. See DataDirect, ODBC driver problems, correcting
message file, Essbase, 25
message file, Integration Server, 25
messages, debug, 15
messages, error, 95
metaoutlines
  access codes, 37
  access permissions, 35
  locks, 32
  opening in exclusive access mode, 32
  opening in standard access mode, 32
  overwriting, 32
  permissions, viewing, 36
  restoring, 64
  tables in OLAP Metadata Catalog, 65
  users, 33
  validating automatically, 23
MO_ACCESS_CODE column, data defined, 37
MO_DESCRIPTIONS table, 66
MO_FILTERS table, 65
MO_INFO table, 36, 66
MO_INTERSECTION table, 67
MO_PROPERTIES table, 66, 67, 68
MOD_ATTRIBUTES table, 66
MOD_INFO table, 66
MODEL_ACCESS_CODE column, data defined, 36
MS_INFO table, 66
MS_PREDICATES table, 66
MS_TX_RULE table, 66
MSR_INFO table, 66
MSR_PREDICATES table, 66
MSR_TX_RULE table, 66
multi-threaded server, 7
multi-user access, 32

N
names, server problems, 54
network listeners, setting, 19
network problems, 42
non-critical error messages, 15
non-Unicode-mode application, defined, 28

O
OA_INFO table, 64
OD_DETAILS table, 64
OD_INFO table, 64
ODBC
common problems, 40
driver compatibility, 58
driver file, specifying on UNIX, 49
environment problems, 44
problems, correcting, 58
testing connection to RDBMS, 45
testing on UNIX, 46
testing on Windows, 45
tracing on UNIX, 61
tracing utility, using, 61
troubleshooting, 39
ODBC Administrator, 45
ODBC Data Sources, odbc.ini file section, 48
odbc.ini file
common problems, 45
inspecting for problems, 48
Trace parameter, 62
odbcconfig utility, 46
ODBCINI variable, 48
OLAP Command Script files, 74
OLAP Metadata Catalog
accessing, 31
backups, 63
connecting to, 88, 89
connection problems, 40, 41
deleting locks, 34
drill-through tables in, 67
Hybrid Analysis tables in, 67
identifying before loading, 78
Integration Server and, 7
manual editing, 63
metaoutline tables, 65
OLAP model tables, 64
problems, database level, 56
recovery, 63
viewing permissions, 36
OLAP models
access codes, 37
access permissions, 35
accessing, 31
displaying primary keys, 23
locks, 32
opening in exclusive access mode, 32
opening in standard access mode, 32
overwriting, 32
permissions, viewing, 35
restoring, 64
tables in OLAP Metadata Catalog, 64
users, 33, 34
OLAPICMD program
problems, 50
starting, 70
olapisvr file, using to start Integration Server, 8
OLAPISVR program
command-line switches, 10
starting in UNIX, 8
starting using startup.bat file, 8
olapisvr.log file, specifying location, 13
OM_DESCRIPTION table, 65
OM_INFO table, 35, 64
OM_PROPERTIES table, 64
OMB_DETAILS table, 65
OMB_INFO table, 65
opening metaoutlines, options, 32
opening OLAP models, options, 32
Oracle
common problems, 41
tnsnames.ora file, 57
orphaned locks, deleting, 33, 34
OV_INFO table, 65
overwriting, OLAP models and metaoutlines, problem, 32
OVL_RELDETAILS table, 65
OVL_RELINFO table, 65
OVP_RELDETAILS table, 65
Index 135

OVP_REL_INFO table, 65

P
parentheses, adding to drill-through filters, 14
password problems, 39, 55
permissions
changing, 35
for metaoutlines, 35, 36
for OLAP models, 35
problems in the RDBMS, 44
shutting down Integration Server, 87
viewing, 35
working with, 31
port 3388, 17, 59, 70
prerequisites
for Integration Server, 7
Integration Services Shell, 69
primary data source, 77
privileges, common problems, 40
problems
starting Integration Server, 7
system-level, 54
TCP/IP, 42
with connections, correcting, 39, 41
with ODBC, correcting, 39, 41, 56
profile file, 47

R
RDBMS
client problems, correcting, 56
common problems, 40
problems, correcting, 56
server name problems, 54
testing connection from database client, 44
testing connection from Integration Server, 49
testing connection from Integration Services
Console, 51
testing connection to ODBC, 45
testing ODBC connection on UNIX, 46
testing ODBC connection on Windows, 45
user account problems, 44
read access, setting, 35
read locks
deleting, 34
viewing, 34
reconnecting to Integration Server, 24
recording data load scripts, 73
recording scripts, 73
records, committing during data load, 11
recovery, OLAP Metadata Catalog, 63
relational database management system. See RDBMS
relational databases
connections to, diagram, 43
problems, correcting, 56
releases
compatibility problems, 50
determining for Integration Services, 76
restoring
metaoutlines, 64
OLAP Metadata Catalogs, 63
OLAP models, 64
restricting access to table records, 18
rights, problems in the RDBMS, 44
round-trip problem, Unicode as solution, 28
running
Integration Server, 8, 24
Integration Services Shell, 69
scripts with Integration Services Shell, 73

S
SAP BW
Hybrid Analysis, 110
support of Integration Services, 29
SAP BW support, 110
scheduling, batch files for loads, 88
scripts
incremental loads, syntax, 75
member and data loads, recording, 73
running with Integration Services Shell, 73
secondary data source, 77
security
access permissions, 35
locks, 32
problems, 42
semicolons, in Integration Services Shell syntax, 72
server problems, 54
server software
port number, 59
problems, correcting, 59
troubleshooting, 39
Servers menu, OLAP Metadata Catalog, Delete Locks
command, 33
SET commands, 71
SETCATALOG command
described, 71, 78
example of, 88, 89
SETLOCALE command, described, 71
SETSOURCE command
described, 71, 77
example of, 88, 89
SETTARGET command
described, 71, 78
example of, 88, 89
setting
data source, 88, 89
Essbase Server, 88, 89
network listeners, 19
OLAP Metadata Catalog, 78, 88, 89
primary data source, 77
secondary data source, 77
target Essbase Server, 78
SHUTDOWN command, 24, 87
shutting down Integration Server, 24, 87
software components
common problem areas, 41
connection problems, 40
source database users, 31
source database, artifact limits, 108
SQL Server, common problems, 41
stack size of thread, specifying, 19
standard access mode, 32
starting
Integration Server, 8, 24
Integration Server, methods, 8
Integration Server, programs necessary, 7
Integration Services Shell, 69
startup switches for Integration Server, 10
startup.bat file, 8, 9
described, 21
storing startup switch settings, 22
using to start Integration Server, 8
STATUS command, described, 72, 76
stopping Integration Server, 24
storage
metaoutline, 63
OLAP model, 63
switches, for OLAPISVR, 10
syntax, Integration Services Shell commands, 72
system-level problems, 54

T
tables
 drill-through, 67
 Hybrid Analysis, 67
 metaoutline, 65
 miscellaneous, 68
 OLAP model, 64
 read-only, 63
 restricting access, 18
 TCP/IP problems, 42
testing
connections, overview, 42
database client, 44
Integration Server connection to Essbase Server, 52
Integration Server connection to RDBMS, 49
Integration Services Console connection to RDBMS, 51
Integration Services Console to Essbase Server, 53
ODBC connection to RDBMS, 45
ODBC on UNIX, 46
ODBC on Windows, 45
threads
specifying number during data load, 16
specifying stack size, 19
timestamp, locks and, 33
tnsnames.ora file, 57
Trace parameter, odbc.ini file, 62
tracing utility
 ODBC for UNIX, 61
 ODBC for Windows, 61
transformation rules, limits with Hybrid Analysis, 111
transformations not supported by Hybrid Analysis, 112
troubleshooting
connections, 39
ODBC, 39
system-level problems, 54
TCP/IP, 42
turning off Integration Server, 24

U
Unicode
 administration tools, 29
 application modes, 27
 features not supported, 27
parameter in LOADALL command, 84, 87
parameter in LOADMEMBER command, 79, 81
Unicode-enabled
    administration tools, 29
    parameter in LOADALL command, 84, 87
    parameter in LOADMEMBER command, 79, 81
Unicode-mode application, defined, 27
UNIX
    ODBC tracing, 61
    odbcconfig utility, 46
    testing ODBC, 46
unload database and application from memory, 20
unreleased locks, deleting, 33, 34
unsupported aggregate functions, 120
user accounts, administrative, 44
user names
    locks and, 33
    problems with, 39, 55
users
    deleting, 31
    disconnecting, 31
    Essbase, defined, 31
    Integration Server, defined, 31
    managing permissions and locks, 31
    multiple, 33
    source database users, 31
UTF-8 encoding, 27

V
configuration parameters, automatic validation, 23
validation of metaoutlines, 23
variables. See environment variables
VERSION command, described, 72, 76
versions, compatibility problems, 50
viewing
    log files, 25
    permissions for metaoutlines, 36
    permissions for OLAP models, 35

W
warning messages, 15
Windows
    ODBC testing, 45
    ODBC tracing, 61
    testing ODBC, 45
    Windows service