## Contents

**Documentation Accessibility** .......................................................... 15

### Chapter 1. Essbase Studio Overview .................................................. 17
- Introducing Essbase Studio ......................................................... 17
- Essbase Studio User Interface .................................................... 17
  - Source Navigator ................................................................. 18
    - Source Navigator: Data Sources Tab ...................................... 18
    - Source Navigator: Minischemas tab ...................................... 18
  - Work Area .............................................................................. 19
  - Metadata Navigator ............................................................. 19
- Logging in to Essbase Studio .......................................................... 20
- Reconnecting to Essbase Studio Server .......................................... 20

### Chapter 2. Administration ........................................................................ 21
- Administration Overview ............................................................. 21
- Compatibility with Essbase Products and Smart View ..................... 21
- Setting Up the Essbase Studio Catalog Database .......................... 21
- Configuring the Server Properties File ......................................... 23
  - Server Properties .................................................................... 24
    - catalog.url ................................................................. 25
    - server.css.URL .......................................................... 26
    - catalog.db ................................................................. 26
    - catalog.username ......................................................... 27
    - catalog.password ........................................................ 27
    - server.hss.bpmApplication ............................................. 27
    - catalog.autoinit .......................................................... 28
    - server.datafile.dir ....................................................... 28
    - server.essbase.streamingCubeBuilding ............................... 28
    - server.timeoutPeriod ..................................................... 29
    - logger.file ................................................................. 29
    - logger.limit ................................................................. 30
    - logger.count ............................................................... 30
## Chapter 5. Minischemas

- Minischema Overview .................................................. 79
- Creating or Editing Minischemas ................................. 80
  - Setting General Properties for Minischemas .................. 80
  - Adding or Removing Tables in a Minischema ................. 81
- Working with Minischemas ........................................... 82
  - Saving a Minischema ................................................. 83
  - Opening a Minischema ............................................... 83
  - Removing Elements from a Minischema ....................... 83
  - Adding or Editing Joins in a Minischema .................... 83
    - Adding or Editing Joins .......................................... 84
    - Adding Joins By Inspection .................................... 86
  - Adding Tables in a Minischema .................................. 86
  - Viewing Sample Data .............................................. 87
  - Applying Color to Minischema Tables ......................... 88
  - Creating Metadata Elements from Minischema Objects ..... 88
  - Viewing Minischemas ............................................... 89
    - Using the Minischema Work Area ............................... 89
    - Maximizing and Minimizing the Minischema Work Area ...... 92
    - Zooming In and Zooming Out on the Minischema Work Area ... 93
  - Arranging the Tables in a Minischema ....................... 93
  - Editing Properties of a Minischema ............................ 93
  - Refreshing the Minischemas List ............................... 94
  - Deleting Minischemas ............................................. 94

## Chapter 6. Introspection

- Introspection Overview ............................................. 95
- Selecting a Minischema Option in the Introspection Wizard ... 96
- Selecting a Fact Table in the Introspection Wizard .......... 96
- Selecting Dimension Tables in the Introspection Wizard .... 97
- Selecting Hierarchies in the Introspection Wizard ........... 97

## Chapter 7. Metadata Elements

- Metadata Elements Overview ..................................... 99
- Creating or Editing Metadata Elements ......................... 100
  - Creating or Editing Dimension Elements and Derived Text Measures 100
  - Creating Dimension Elements .................................. 100
  - Defining or Editing General Properties for Dimension Elements 102
  - Creating or Editing Derived Text Measures .................. 107
  - Creating or Editing Alias Set Bindings for a Given Alias Set 108
<table>
<thead>
<tr>
<th>Chapter 10. Cube Schemas</th>
<th>137</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cube Schema Overview</td>
<td>137</td>
</tr>
<tr>
<td>Creating or Editing Cube Schemas</td>
<td>137</td>
</tr>
<tr>
<td>Choosing Measures and Hierarchies for a Cube Schema</td>
<td>138</td>
</tr>
<tr>
<td>Previewing Hierarchies</td>
<td>139</td>
</tr>
<tr>
<td>Setting Cube Schema Options</td>
<td>139</td>
</tr>
<tr>
<td>Defining Data Load Mappings</td>
<td>140</td>
</tr>
<tr>
<td>Creating Essbase Models from Existing Cube Schemas</td>
<td>142</td>
</tr>
<tr>
<td>Viewing Deployment History</td>
<td>142</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 11. Essbase Properties</th>
<th>145</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model Properties</td>
<td>145</td>
</tr>
<tr>
<td>About Essbase Models</td>
<td>145</td>
</tr>
<tr>
<td>Models Overview</td>
<td>146</td>
</tr>
<tr>
<td>Accessing the Model Properties Dialog Box</td>
<td>147</td>
</tr>
<tr>
<td>Setting General Model Properties</td>
<td>147</td>
</tr>
<tr>
<td>Overriding Standard Data Load SQL</td>
<td>148</td>
</tr>
<tr>
<td>Data Load SQL Override Editing Guidelines</td>
<td>149</td>
</tr>
<tr>
<td>OLAP Overview</td>
<td>150</td>
</tr>
<tr>
<td>XOLAP Overview</td>
<td>151</td>
</tr>
<tr>
<td>Designating a Model for XOLAP</td>
<td>152</td>
</tr>
<tr>
<td>Selecting Tables as Alias Sets</td>
<td>153</td>
</tr>
<tr>
<td>Defining Attributes in Models</td>
<td>153</td>
</tr>
<tr>
<td>Attributes Overview</td>
<td>154</td>
</tr>
<tr>
<td>Varying Attributes Overview</td>
<td>155</td>
</tr>
<tr>
<td>Setting Attribute Member Names Format</td>
<td>156</td>
</tr>
<tr>
<td>Setting Attribute Calculations Member Names Format</td>
<td>157</td>
</tr>
<tr>
<td>Specifying Attribute Boolean, Date, and Numeric Ranges</td>
<td>158</td>
</tr>
<tr>
<td>Dimension Properties</td>
<td>159</td>
</tr>
<tr>
<td>Dimensions Overview</td>
<td>160</td>
</tr>
<tr>
<td>Dimension Types</td>
<td>160</td>
</tr>
<tr>
<td>Rules for Using Time Dimensions</td>
<td>161</td>
</tr>
<tr>
<td>Rules for Using Accounts Dimensions</td>
<td>161</td>
</tr>
<tr>
<td>Accessing the Dimension Properties Tabs</td>
<td>161</td>
</tr>
<tr>
<td>Setting General Dimension Properties</td>
<td>162</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Editing Dimensions</td>
<td>163</td>
</tr>
<tr>
<td>Selecting the Dimension Type</td>
<td>163</td>
</tr>
<tr>
<td>Using Dynamic Time Series</td>
<td>164</td>
</tr>
<tr>
<td>Selecting the Dimension Storage Method</td>
<td>165</td>
</tr>
<tr>
<td>Selecting Two Pass Calculation</td>
<td>165</td>
</tr>
<tr>
<td>Two Pass Calculation Overview</td>
<td>166</td>
</tr>
<tr>
<td>Selecting the Data Storage Method</td>
<td>166</td>
</tr>
<tr>
<td>Selecting the Solve Order</td>
<td>167</td>
</tr>
<tr>
<td>Editing Account Dimensions Properties</td>
<td>168</td>
</tr>
<tr>
<td>Selecting a Time Balance</td>
<td>169</td>
</tr>
<tr>
<td>Time Balance Overview</td>
<td>169</td>
</tr>
<tr>
<td>Selecting the Variance Reporting Method</td>
<td>170</td>
</tr>
<tr>
<td>Selecting the Skip Option</td>
<td>170</td>
</tr>
<tr>
<td>Adding Formulas to Dimensions</td>
<td>171</td>
</tr>
<tr>
<td>Formulas Overview</td>
<td>172</td>
</tr>
<tr>
<td>Displaying Dimension Aliases</td>
<td>172</td>
</tr>
<tr>
<td>Assigning User-Defined Attributes to Dimensions</td>
<td>173</td>
</tr>
<tr>
<td>UDAS Overview</td>
<td>173</td>
</tr>
<tr>
<td>Selecting Outline Build Options</td>
<td>174</td>
</tr>
<tr>
<td>Selecting Alternate Hierarchy Storage Settings</td>
<td>174</td>
</tr>
<tr>
<td>Moving Duplicate Member Settings</td>
<td>175</td>
</tr>
<tr>
<td>Optimizing Data Loads</td>
<td>175</td>
</tr>
<tr>
<td>Member Properties</td>
<td>176</td>
</tr>
<tr>
<td>Members Overview</td>
<td>176</td>
</tr>
<tr>
<td>Accessing the Member Properties Tabs</td>
<td>176</td>
</tr>
<tr>
<td>Setting General Member Properties</td>
<td>177</td>
</tr>
<tr>
<td>Specifying General Member Properties</td>
<td>177</td>
</tr>
<tr>
<td>Selecting Members as Attributes</td>
<td>178</td>
</tr>
<tr>
<td>Setting Varying Attributes for Members</td>
<td>178</td>
</tr>
<tr>
<td>Editing Member Information</td>
<td>182</td>
</tr>
<tr>
<td>Selecting Consolidation Methods for Children Members</td>
<td>182</td>
</tr>
<tr>
<td>Consolidation of Children Members Overview</td>
<td>183</td>
</tr>
<tr>
<td>Selecting the Member Calculation Solve Order</td>
<td>183</td>
</tr>
<tr>
<td>Solve Order Overview</td>
<td>184</td>
</tr>
<tr>
<td>Selecting a Two Pass Calculation Option</td>
<td>185</td>
</tr>
<tr>
<td>Selecting the Member Data Storage Method</td>
<td>185</td>
</tr>
<tr>
<td>Aggregate Storage with Label Only Option</td>
<td>186</td>
</tr>
<tr>
<td>Selecting an Aggregate Storage Option</td>
<td>186</td>
</tr>
<tr>
<td>Aggregate Storage Guidelines</td>
<td>187</td>
</tr>
<tr>
<td>Topic</td>
<td>Page</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Editing Members in Accounts Dimensions</td>
<td>188</td>
</tr>
<tr>
<td>Selecting the Time Balance</td>
<td>189</td>
</tr>
<tr>
<td>Selecting a Skip Option</td>
<td>189</td>
</tr>
<tr>
<td>Selecting a Variance Reporting Method</td>
<td>190</td>
</tr>
<tr>
<td>Specifying Data Load Scaling</td>
<td>191</td>
</tr>
<tr>
<td>Adding Formulas to Members</td>
<td>192</td>
</tr>
<tr>
<td>Editing Member Aliases</td>
<td>192</td>
</tr>
<tr>
<td>Aliases Overview</td>
<td>193</td>
</tr>
<tr>
<td>Displaying Alias Sets</td>
<td>194</td>
</tr>
<tr>
<td>Changing and Reformattting Alias Names</td>
<td>194</td>
</tr>
<tr>
<td>Creating a Search Rule to Change and Reformatt Alias Names</td>
<td>194</td>
</tr>
<tr>
<td>Modifying a Search Rule to Change and Reformatt Alias Names</td>
<td>195</td>
</tr>
<tr>
<td>Changing Cases in Aliases</td>
<td>196</td>
</tr>
<tr>
<td>Reformating Spaces in Aliases</td>
<td>197</td>
</tr>
<tr>
<td>Adding Prefixes to Aliases</td>
<td>197</td>
</tr>
<tr>
<td>Adding Suffixes to Aliases</td>
<td>199</td>
</tr>
<tr>
<td>Assigning User-Defined Attributes to Members</td>
<td>200</td>
</tr>
<tr>
<td>Transforming Members</td>
<td>200</td>
</tr>
<tr>
<td>Member Name Transformation Sequence</td>
<td>201</td>
</tr>
<tr>
<td>Adding Prefixes to Members</td>
<td>201</td>
</tr>
<tr>
<td>Adding Suffixes to Members</td>
<td>202</td>
</tr>
<tr>
<td>Changing and Reformattting Member Names</td>
<td>204</td>
</tr>
<tr>
<td>Creating a Search Rule to Change and Reformatt Member Names</td>
<td>204</td>
</tr>
<tr>
<td>Modifying a Search Rule to Change and Reformatt Member Names</td>
<td>205</td>
</tr>
<tr>
<td>Changing Cases in Members</td>
<td>206</td>
</tr>
<tr>
<td>Reformating Spaces in Members</td>
<td>207</td>
</tr>
<tr>
<td>Working with Essbase Model Properties</td>
<td>207</td>
</tr>
<tr>
<td>Viewing Models</td>
<td>207</td>
</tr>
<tr>
<td>Opening the Model Work Area</td>
<td>208</td>
</tr>
<tr>
<td>Using the Model Work Area</td>
<td>208</td>
</tr>
<tr>
<td>Validating Model Properties</td>
<td>209</td>
</tr>
<tr>
<td>Reviewing Changes to Properties</td>
<td>210</td>
</tr>
<tr>
<td>Browsing Models</td>
<td>210</td>
</tr>
<tr>
<td>Using Tool Tips</td>
<td>211</td>
</tr>
<tr>
<td>Chapter 12. Cube Deployment</td>
<td>213</td>
</tr>
<tr>
<td>Cube Deployment Overview</td>
<td>213</td>
</tr>
<tr>
<td>Deploying Cubes</td>
<td>214</td>
</tr>
<tr>
<td>Creating a Connection to Essbase Server</td>
<td>214</td>
</tr>
</tbody>
</table>
Chapter 13. Deployed Applications

Chapter 14. Lineage

Chapter 15. Drill-through Reports

Chapter 16. Find and Search

Chapter 17. Preferences
dayOfMonth ..................................................... 285
month .......................................................... 286
monthName .................................................... 286
monthShortName ............................................ 286
quarter .......................................................... 287
quarterAsString .............................................. 287
weekday .......................................................... 288
weekdayName ................................................. 288
weekdayShortName ............................................. 288
year ............................................................. 289
yearShort ...................................................... 289
String ............................................................ 290
contains .......................................................... 290
index .................................................................. 290
leftStr ............................................................. 291
length .............................................................. 291
lower ............................................................... 291
lTrim ............................................................... 292
rightStr ........................................................... 292
rTrim ............................................................... 293
soundex ............................................................ 293
substr .............................................................. 293
trim ................................................................. 294
upper ............................................................... 294
Numeric ............................................................ 295
abs ................................................................. 295
exp .................................................................. 295
ln ..................................................................... 295
log10 ............................................................... 296
pow ................................................................. 296
sqrt ................................................................ 297
Glossary ............................................................ 299
Index ............................................................... 323
Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at http://www.oracle.com/accessibility/.

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, seven days a week. For TTY support, call 800.446.2398.
Introducing Essbase Studio

Oracle Essbase Studio simplifies cube construction by delivering a single environment for performing tasks related to data modeling, cube designing, and analytic application construction. By consolidating cube construction activities into one interface, Essbase Studio provides a consistent platform for building outlines and loading data.

With a wizard-driven user interface, Essbase Studio supports modeling various data source types from which Oracle Essbase applications are typically built, making it a single point from which all cube-related data modeling can be performed.

A common metadata repository, or catalog, captures all metadata related to all Essbase applications built in the enterprise and allows the reuse of metadata at the lowest level of granularity. The catalog gives Essbase Studio knowledge of the common metadata that is shared across the various applications enterprise-wide.

Essbase Studio supports several drill-through options: relational databases, Oracle Business Intelligence Enterprise Edition, URLs, custom SQL, and Java methods. Drill-through functionality is supported from data cells and member cells and is dynamically linked to cubes with matching metadata context.

Essbase Studio also supports lineage tracking through a rich graphical view of the metadata relationships, allowing users to follow application lineages to their metadata components and through to the data sources from which they were sourced.

Essbase Studio User Interface

The Essbase Studio user interface consists of three dockable main areas:

- Source Navigator, described in “Source Navigator” on page 18
- Work Area, described in “Work Area” on page 19
Source Navigator

The Source Navigator, displayed by default in the right pane of the Essbase Studio Console, has two tabs, the Data Sources tab and the Minischemas tab.

- **Data Sources tab**—Lists the physical data sources to which you have created connections. You can also launch the Connection Wizard from this tab, where you create data source connections.

  For more information, see “Source Navigator: Data Sources Tab” on page 18.

- **Minischemas tab**—Lists the graphical representations of the tables you select from one or more data sources connections. You can create minischemas when creating data source connections, or you can create them later.

  For more information, see “Source Navigator: Minischemas tab” on page 18.

Source Navigator: Data Sources Tab

The Data Sources tab lists the physical data sources to which you have created connections. You can also launch the Connection Wizard from this tab, where you create data source connections. Other tasks that you can perform from the Data Sources tab:

- View sample data from a selected table in a data source
- View basic properties of data sources, tables, and columns
- Show the “friendly names” of tables in a Microsoft SQL Server data source
- Delete data source connections
- Refresh the data sources list
- Introspection, an analysis of the data source connection to identify possible hierarchies
- Incrementally update an existing data source
- Create user-defined tables

See Chapter 4, “Data Source Connections” for more information.

Source Navigator: Minischemas tab

The Minischemas tab lists the minischemas you have created. Minischemas are graphical representations of the data sources to which you have created connections. You can create minischemas when creating data source connections, or you can create them later. Minischemas may contain a subset of the tables in a data source connection or all the tables. Alternatively, minischemas can contain tables from multiple data sources.

Other tasks that you can perform from the Minischemas tab:

- View properties of the minischema, including source, table and column properties
● View sample data from tables in a physical data source
● Edit the minischema to add or remove tables from one or more data source connections
● Add or edit joins
● Delete minischemas
● Refresh the minischemas list

For more information, see Chapter 5, “Minischemas.”

**Work Area**

The work area, by default in the middle pane of the Essbase Studio Console, is used to display and work with metadata elements and graphical representations of source and metadata elements.

Objects displayed in the work area:

● Minischemas, described in “Minischema Overview” on page 79
● Essbase models, described in Chapter 11, “Essbase Properties”
● Lineage View, described in Chapter 14, “Lineage”
● Drill-through reports, described in Chapter 15, “Drill-through Reports”
● Hierarchy editor, described in Chapter 9, “Hierarchies”
● Sample data, described in the topic, “Viewing Sample Data” on page 75 in Chapter 4, “Data Source Connections”
● Deployment history, described in “Viewing Deployment History” on page 219

**Metadata Navigator**

The Metadata Navigator, displayed by default in the left pane of the Essbase Studio Console, contains the following:

● Metadata elements derived from the physical data sources when you create a data source connection.

  Each time you create a new data source connection, you can create metadata elements from the physical elements in the data source to which you have connected. These metadata elements are displayed by default in a folder structure that mimics the tables and columns in the data source.

  The Metadata Navigator displays metadata elements from multiple data source connections as well as objects from any type of supported data source. For example, your business may require access to data from sources as varied as relational, text file data, Oracle Business Intelligence, and Oracle Hyperion EPM Architect, Fusion Edition. After you create the connections to these sources, the metadata elements derived from each data source are displayed in a tree structure in the Metadata Navigator. Default folders are created to contain the metadata elements for each data source and, optionally, you can create folders during the data source creation process to further organize the metadata elements in the tree.
- Metadata elements that you create from the derived metadata artifacts in the Metadata Navigator.

From the derived metadata artifacts you can further create metadata elements such as:

- Folders
- Dimension elements
- Derived text measures
- Standard hierarchies
- Measure hierarchies
- Calendar hierarchies
- Cube schemas
- Essbase models
- Drill-through reports

These metadata elements are used to create cube schemas and models that work with Essbase. Essbase models are used to create and deploy Essbase cubes.

Logging in to Essbase Studio

When you start Essbase Studio, the Login dialog box is displayed.

➢ To log in to Essbase Studio:

1. In Server, provide the name of the computer on which Essbase Studio server is installed; for example: aspen3

2. Provide your User name and Password, then click Log On.

You are now ready to begin work with Essbase Studio.

Reconnecting to Essbase Studio Server

The connection between Essbase Studio Server and Essbase Studio Console can be lost if the console is idle for more than one hour (or the length of time you specify, as described in Chapter 2, “Administration”).

➢ To restore the connection between Essbase Studio Server and Essbase Studio Console:

1. In the Essbase Studio Console, select Tools > Reconnect to Server to display the Login dialog box.
   The Server and User fields are read only.

2. Enter your Password, then click Log On.
Administration Overview

When you install Essbase Studio using the Oracle Hyperion Enterprise Performance Management System Installer, Fusion Edition, most configuration tasks are done for you as part of the configuration process. Oracle recommends that, where possible, you always use the Oracle’s Hyperion Enterprise Performance Management System Configurator whenever you need make changes your configuration. You may, however, manually edit some of the configuration files.

Compatibility with Essbase Products and Smart View

Essbase Studio works only with the 11.1.1 versions of Essbase products. Before you begin working with Essbase Studio, ensure that you have installed and configured the 11.1.1 versions of Essbase Server, Essbase Administration Server and Console, Oracle Hyperion Provider Services, and Oracle Hyperion Smart View for Office, Fusion Edition.

Setting Up the Essbase Studio Catalog Database

Before you begin working with Essbase Studio, you must decide where you will set up your Essbase Studio catalog database (sometimes referred to as “the catalog”). You may include the catalog database in one database instance that serves all Hyperion products, or you may create a dedicated relational database on your database server computer expressly for the catalog.
The catalog is the metadata repository for Essbase Studio. When Essbase Studio users create any kind of metadata element in Essbase Studio—for example, dimension elements, hierarchies, cube schemas, and Essbase models—the metadata for those elements is stored in the Essbase Studio catalog.

During the configuration process (after installation), you use the EPM System Configurator to configure Essbase Studio to use the catalog database that you specify, whether it is part of one database instance for all Hyperion products, or a database expressly created to hold the Essbase Studio catalog.

Information about the catalog database is stored in server.properties, located in the Essbase Studio server directory. It contains the name, location, user name, and the encrypted password for this catalog database. When you start Essbase Studio, server.properties is read to determine the catalog database location and security credentials.

You can use any supported relational database to hold the Essbase Studio catalog database.

Notes:

- You must specify a catalog database user during the configuration process. The catalog database user must have at least write privileges to the Essbase Studio catalog database.
- Each Essbase Studio Server instance must have its own catalog database.
- If Oracle is your catalog database: Each time you start Essbase Studio Server, the server will try to execute the following ALTER SYSTEM statement:

  ALTER SYSTEM SET open_cursors=300 SCOPE=MEMORY

  If the catalog user (specified in the server.properties file during configuration) does not have the privileges necessary to execute the ALTER SYSTEM statement, Essbase Studio will log a warning message and continue working.

If you are using one database instance for all Hyperion products, refer to the Oracle's Hyperion® Shared Services documentation for information on setting up this database.

If you are using a dedicated database for your catalog, complete the following procedure.

➤ To set up a dedicated catalog database for Essbase Studio:

1. In your RDBMS, create an empty database schema.

   This will be your Essbase Studio catalog database.

   After configuration, this is the database specified in the catalog.db property in the server.properties file.

2. In your RDBMS, grant at least write privileges to the user who will be the designated as the database user of the catalog.

   After configuration, this is the user specified in the catalog.username and catalog.password properties in the server.properties file. This must be a user with at least write privileges to the database specified in the catalog.db property. These properties are set during the configuration process.
If you are using the Shared Services repository for the catalog database, the same privilege requirements apply.

3 After Essbase Studio installation, run the EPM System Configurator and provide the information requested for the catalog database.

4 After configuration, in `<essbase_studio>\server\server.properties`, verify the information related to the catalog database:
   - `catalog.db`—The name of the catalog database you created in step 1.

   **Note:**
   If you configured the catalog database for an IBM DB2 or Microsoft SQL Server database, the schema name should be appended to the name of the database; for example, `catalog.db=esbstudio.dbo` where `esbstudio` is the database name, and `dbo` is the schema name.
   - `catalog.username`—A user with at least write privileges for the catalog database. See “catalog.username” on page 27.
   - `catalog.password`—The encrypted password for the catalog database user. See “catalog.password” on page 27.

When finished, close `server.properties`

Essbase Studio Server reads the `server.properties` file at startup for this catalog database information. See “Configuring the Server Properties File” on page 23 for information on configuration options.

---

## Configuring the Server Properties File

A default `server.properties` file, created during installation, contains the server properties necessary to run Essbase Studio.

During configuration, the EPM System Configurator writes the server configuration settings you select to the Essbase Studio `server.properties` file.

After installation, you may can edit or add server properties.

**Note:**

The `server.properties` file is located in `HYPERION_HOME/products/Essbase/EssbaseStudio/Server`, where by default, `HYPERION_HOME` is `C:/Hyperion`.

> To edit the `server.properties` file:

1. Edit the file in text format with any text editor, such as Windows Notepad.

2. Enter each setting on a separate line in the file.
   
   You need not end each line with a semicolon.

3. Ensure that the file is named `server.properties`.

---
4 Save the file in the Essbase Studio server directory.

5 After changing the server properties file, stop and restart Essbase Studio Server.

Essbase Studio Server reads the properties file once, at startup.

The following is an example of the properties that can be configured in server.properties. See “Server Properties” on page 24 for descriptions and examples of each property.

catalog.url=database tag://hostname:port/
server.css.URL=server.css.URL=http://server.us.oracle.com:port/interop/
catalog.db=catalog database name
catalog.username=catalog user ID
catalog.password=encrypted catalog password
server.hss.bpmApplication=BPM:BPM application ID
catalog.autoinit=true/false
server.datafile.dir=path to flat file directory
server.essbase.streamingCubeBuilding=true/false
server.timeoutPeriod=number of seconds
logger.file=name of server log file
logger.limit=maximum size of server log files in bytes
logger.count=number of server log files in log history
server.queueSize=size of the task queue
server.threadCount=number of worker threads
server.resourceCount=number of tasks that can be executed concurrently
server.sql.fetchSize=fetch size
server.tempDir=path to temporary directory
server.charset=charset
server.readLockTimeOut=number of seconds
server.writeLockTimeOut=number of seconds
com.hyperion.cp.level=logging level
server.essbase.TPTapi=true/false
server.essbase.disableDistinct=true/false

Note:

Some exceptions to this syntax exist, depending on your RDBMS. They are discussed in “Server Properties File Examples” on page 35.

Server Properties

Following are the configurable server properties:

- “catalog.url” on page 25
- “server.css.URL” on page 26
- “catalog.db” on page 26
- “catalog.username” on page 27
- “catalog.password” on page 27
- “server.hss.bpmApplication” on page 27
- “catalog.autoinit” on page 28
- “server.datafile.dir” on page 28
catalog.url

The URL for the catalog database.

Syntax

database_tag://hostname:port/

Notes

The following database tags are supported:

- oracle—Oracle
- db2—IBM DB2
- sqlserver—Microsoft SQL Server
- mysql—MySQL

When you are using an Oracle database as your catalog database, you must append the Oracle system ID (SID) to the catalog.url parameter, using the following syntax:

catalog.url=oracle://hostname.domain name.com:port\OracleSID

When you are using an IBM DB2 database as your catalog database, you must append the database name to the catalog.url parameter, using the following syntax:

database tag://hostname:port;databaseName=database name
Example

catalog.url=sqlserver://spruce:3307

See “Server Properties File Examples” on page 35 for specific RDBMS examples.

**server.css.URL**

The URL for the Shared Services server.

**Syntax**

server.css.URL=http://hostname.domain name.com:port/interop/framework/getCSSConfigFile

**Example**


**catalog.db**

The name of the relational database that has been set up to be the metadata repository (also known as the “catalog database” or “catalog”) for Essbase Studio.

**Syntax**

catalog.db=catalog database name

**Note:**

If the catalog database name or schema name starts with a number, you must place quotation marks (" ") around the database name when setting the catalog.db property.

**Example**

catalog.db=esbstudio

IBM DB2 and Microsoft SQL Server users must include the schema name in the catalog.db parameter, using the following syntax:

catalog.db=catalog database name.schema name

If your catalog database name or schema name starts with a number, be sure to place quotation marks (" ") around the database name. Below are examples of correct and incorrect definitions for catalog database names.

Incorrect—catalog.db=123a

Correct—catalog.db="123a"

Incorrect—catalog.db=123a.user1

Correct—catalog.db="123a".user1
Incorrect—`catalog.db=a123.1user`
Correct—`catalog.db=a123."1user"

**catalog.username**
The user ID for a user of the catalog database. This user must have at least write privileges to the catalog.

**Syntax**

```
catalog.username=catalog user ID
```

**Example**

```
catalog.username=root
```

**catalog.password**
The encrypted password for the user specified in `catalog.username`.

**Note:**
The encrypted password string is generated by the EPM System Configurator. If you edit this property, you must use an encrypted password string.

**Syntax**

```
catalog.password=encrypted catalog password
```

**Example**

```
catalog.password=A627FC9A6DEA834C1FA777217871D09E
```

**server.hss.bpmApplication**
The application identification number assigned to Essbase Studio Server by Shared Services during the Oracle Hyperion Enterprise Performance Management System configuration process.

The EPM System Configurator sets this property automatically during registration of Essbase Studio Server. To change the `server.hss.bpmApplication` property to point to a different instance of Shared Services, you must run the EPM System Configurator and register the new Shared Services instance.

**Syntax**

```
server.hss.bpmApplication=BPM\:application identifier
```

**Example**

```
server.hss.bpmApplication=BPM\:29696
```
**catalog.autoinit**

When set to “true,” automatically initializes the catalog database when restarting the Essbase Studio Server after the catalog database has been cleared.

The default value is true.

**Syntax**

catalog.autoinit=true|false

**Example**

catalog.autoinit=false

**server.datafile.dir**

Defines the root directory for flat files that will serve as a data source for Essbase Studio Server.

The default value is ./flatfiles.

**Syntax**

server.datafile.dir=path to flat file directory

**Notes**

You can override the default directory by specifying the full path to the flat files location. For example:

C:\\Hyperion\\EssbaseStudio\\server\\samples\\tbc_txt

Note that the backslash character is a special character and, therefore, must be preceded by a backslash character (an Escape sequence).

**Note:**

If server.datafile.dir is changed, then Essbase Studio Server must be restarted for the modification to take effect.

**Example**

server.datafile.dir=$USER_HOME/data/FlatFileDir

**server.essbase.streamingCubeBuilding**

When set to true, Essbase cube deployment occurs in streaming mode.

The default value is false, meaning Essbase Studio is run in nonstreaming mode.

**Syntax**

server.essbase.streamingCubeBuilding=true|false
Notes
Nonstreaming mode means that during cube deployment, Essbase Studio Server queries the external data source using an ODBC connection.

Streaming mode means that during cube deployment, Essbase Studio Server queries the external data source directly.

Example
server.essbase.streamingCubeBuilding=true

server.timeoutPeriod
The amount of time, in seconds, that Essbase Studio Console can remain idle before losing its connection to Essbase Studio Server.
The default value is 3600 seconds (1 hour).

Syntax
server.timeoutPeriod=number of seconds

Example
server.timeoutPeriod=7200

logger.file
The name of the server log file.
The default server log file name is server.log.
The log file is located by default in:
HYPERION_HOME/logs/esbstudio
You cannot change the log file location.

Syntax
logger.file=name of server log file

Notes on logger.file, logger.limit, and logger.count
The name of the current log file corresponds to the name you specify for the logger.file property. If this parameter is not specified, the default server log file name, server.log, is used.

Log history files are older versions of the log file that are created automatically when the size of the server log file reaches the limit specified in the logger.limit parameter. The extension “.n” is automatically appended to the log history files; for example, server.log.1, server.log.2, and so on. When the size of the current log file becomes bigger than the size specified in the logger.limit parameter, the current log file and all log history files are incrementally renamed, and a new current log file is created. When the number of log history
files reaches the limit set in the `logger.count` parameter, the oldest of the log history files is removed to accommodate the new log file being created.

For example, by default, when the server log file reaches the 10 MB limit, the following happens automatically: `server.log` becomes `server.log.1`, `server.log.1` becomes `server.log.2`, `server.log.2` becomes `server.log.3`, and so forth; and Essbase Studio Server creates the new current log file, `server.log`.

**Example**

```
logger.file=server_prod.log
```

### `logger.limit`

The maximum size of the server log files in bytes.

The default server log file size is 10485760 bytes (10 MB).

This property works together with the “`logger.file` on page 29” and “`logger.count` on page 30” properties. See Notes on `logger.file`, `logger.limit`, and `logger.count` in “`logger.file` on page 29” for more information.

**Syntax**

```
logger.limit=maximum size of server log files in bytes
```

**Example**

```
logger.limit=15000000
```

### `logger.count`

The number of server log files that are stored as log history files.

The default number of server log files that are kept as log history files is 10.

This property works together with the “`logger.file` on page 29” and “`logger.limit` on page 30” properties. See Notes on `logger.file`, `logger.limit`, and `logger.count` in “`logger.file` on page 29” for more information.

**Syntax**

```
logger.count=number of server log files in log history
```

**Example**

```
logger.count=15
```

### `server.queueSize`

Sets the queue size in terms of number of tasks. The queue holds the tasks that are waiting to be executed by Essbase Studio Server.
The default queue size is 200 tasks.

Syntax

server.queueSize=size of the task queue

Example

server.queueSize=250

**server.threadCount**

The number of worker threads allocated to Essbase Studio Server.

Syntax

server.threadCount=number of worker threads

Notes

Worker threads are:

- Listeners; threads that listen for requests from Essbase Studio clients
- Threads that get tasks from the clients
- The tasks themselves; the tasks to be executed by Essbase Studio Server

The default is 30 threads.

Note:

To account for the worker threads that listen for and get tasks from Essbase Studio clients, the number of workers threads must be set to a number greater than the number of resources set in “server.resourceCount” on page 31.

Example

server.threadCount=40

**server.resourceCount**

Sets the maximum number of resources. The number of resources defines the number of tasks that Essbase Studio Server can execute concurrently.

The default is 20 resources.

Syntax

server.resourceCount=number of tasks that can be executed concurrently

Example

server.resourceCount=35
**server.sql.fetchSize**

Redefines the default size, in number of records, of the JDBC driver fetch buffer.

The default buffer size Essbase Studio Server is 1000 records.

**Syntax**

```
server.sql.fetchSize=fetch size in number of records
```

**Example**

```
server.sql.fetchSize=500
```

**server.tempDir**

Specifies the directory for temporary files created by Essbase Studio Server, such as rules files and error files. This directory must exist.

The default value defines the relative path to the directory `.\ess_japihome\data`, created during Essbase Studio installation.

**Syntax**

```
server.tempDir=path to temporary directory
```

**Example**

```
server.tempDir=$USER_HOME/Temp
```

**server.charset**

Specifies the character set that Essbase Studio Server uses for conversion of all messages that are placed in the server log file.

The default character set is `utf-8`.

**Syntax**

```
server.charset=charset
```

**Example**

```
server.charset=US-ASCII
```

**server.readLockTimeOut**

Specifies the number of seconds before timing out that a process will wait when making a request to read information from the Essbase Studio catalog database.

The default is 120 seconds.

This property works together with the `server.writeLockTimeOut` property, described in “server.writeLockTimeOut” on page 33.
Syntax

server.readLockTimeOut=number of seconds

Notes
When more than one user is accessing the same Essbase Studio catalog database, certain user actions can block other users from accessing that catalog database. These user actions are:

● Creating metadata elements
● Modifying metadata elements
● Data exploration (creating a data source connection, including table selection, metadata element creation; also, incremental data source update)

When a user performs one of the actions listed above, Essbase Studio Server blocks other users' requests to the same catalog database for the time period set in `server.readLockTimeOut`. If the time period passes and the server is unable to fulfill the request, a message is displayed informing the user that the processing of the request has been interrupted.

In some cases, if Essbase Studio can fulfill a request by fetching an object from cache, it will attempt to do so. For example, if a user selects a hierarchy to view, and that hierarchy is present in the cache, Essbase Studio will display it to the user.

Example

server.readLockTimeOut=90

server.writeLockTimeOut

Specifies the number of seconds before timing out that a process will wait when making a request to write to the Essbase Studio catalog database.

The default is 120 seconds.

This property works together with the `server.readLockTimeOut` property, described in “server.readLockTimeOut” on page 32.

Syntax

server.writeLockTimeOut=number of seconds

Notes
See the Notes section in “server.readLockTimeOut” on page 32.

Example

server.writeLockTimeOut=150

com.hyperion.cp.level

Specifies the logging level (the level of information written to the log file specified in the `logger.file` property).
The default is INFO.

**Syntax**

```java
com.hyperion.cp.level=logging level
```

Available logging levels:
- **SEVERE**—The highest level. Includes messages that indicate serious failures.
- **WARNING**—Includes messages that indicate potential problems.
- **INFO**—Includes information messages.
- **CONFIG**—Includes static configuration messages.
- **FINE**—Includes tracing information.
- **FINER**—Includes fairly detailed tracing messages.
- **FINEST**—The lowest level. Includes highly detailed tracing messages.
- **OFF**—Turns off logging.
- **ALL**—Enables logging of all messages.

**Example**

```java
com.hyperion.cp.level=FINEST
```

**server.essbase.TPTapi**

For Teradata users only.

When set to “true,” enables the Teradata Parallel Transporter API, which results in faster data load performance compared to an ODBC connection.

The default value is false.

For more information about Teradata Parallel Transporter, see the *Oracle Essbase SQL Interface Guide*.

**Syntax**

```java
server.essbase.TPTapi=true|false
```

**Example**

```java
server.essbase.TPTapi=true
```

**server.essbase.disableDistinct**

When set to “true,” allows users to disable the DISTINCT filter in member load queries.

The default value is false.

**Syntax**

```java
server.essbase.disableDistinct=true|false
```
Notes

By default, when performing a member load, Essbase Studio Server adds the DISTINCT keyword to filter out duplicate records.

Example

server.essbase.disableDistinct=true

Server Properties File Examples

The following examples are presented:

- “Oracle Example” on page 35
- “IBM DB2 Example” on page 36
- “Microsoft SQL Server Example” on page 36
- “MySQL Example” on page 37

Oracle Example

When you use an Oracle database as your catalog database, you must append the Oracle system ID (SID) to the catalog.url parameter, using the following syntax:

catalog.url=oracle://hostname.domain.com:port\OracleSID

The following is an example configuration for the server.properties file using an Oracle database:

catalog.url=oracle://sequoia.xyzco.com:1521\bpm
catalog.db=esbstudio
catalog.username=root
catalog.password=A627FC9A6DEA834C1FA777217871D09E
server.hss.bpmApplication=BPM\:29696
catalog.autoinit=true
server.datafile.dir=./data/flatfiles
server.essbase.streamingCubeBuilding=true
server.timeoutPeriod=7200
logger.file=server_prod.log
logger.limit=15000000
logger.count=5
server.queueSize=250
server.threadCount=40
server.resourceCount=30
server.sql.fetchSize=100
server.tempDir=C:/Windows/Temp
server.charset=windows-1251
server.readLockTimeOut=90
server.writeLockTimeOut=150
com.hyperion.cp.level=FINEST
**IBM DB2 Example**

When you use a DB2 database as your catalog database, you must append the database name to the catalog.url parameter, using the following syntax:

```
database tag://hostname:port;databaseName=database name
```

DB2 users must also include the schema name in the catalog.db parameter, using the following syntax:

```
catalog.db=catalog database name.schema name
```

The following is an example configuration for server.properties using a DB2 database:

```properties
catalog.url=db2://cypress:50000;databaseName=BPM_Db
catalog.db=bpm.ROOT
catalog.username=root
catalog.password=A627FC9A6DEA834C1FA777217871D09E
server.hss.bpmApplication=BPM:29696
catalog.autoinit=true
server.datafile.dir=${USER_HOME}/data/FlatFileDir
server.essbase.streamingCubeBuilding=true
server.timeoutPeriod=7200
logger.file=server_test.log
logger.limit=20971520
logger.count=15
server.queueSize=250
server.threadCount=50
server.resourceCount=35
server.sql.fetchSize=500
server.tempDir=${USER_HOME}/Temp
server.readLockTimeOut=150
server.writeLockTimeOut=150
com.hyperion.cp.level=WARNING
```

**Microsoft SQL Server Example**

Microsoft SQL Server users must include the schema name in the catalog.db parameter, using the following syntax:

```
catalog.db=catalog database name.schema name
```

The following is an example configuration for server.properties using a SQL Server database:

```properties
catalog.url=sqlserver://aspen3.us.oracle.com:1433
catalog.db=esbstudio.root
catalog.username=root
catalog.password=A627FC9A6DEA834C1FA777217871D09E
server.hss.bpmApplication=BPM:29696
catalog.autoinit=true
server.datafile.dir=C:/FlatFileSource
server.essbase.streamingCubeBuilding=true
server.timeoutPeriod=7200
```
MySQL Example

The following is an example configuration for server.properties using a MySQL database:

catalog.url=mysql://maple5.xyzco.com:3307/
getCSSConfigFile
catalog.db=esbstudio
catalog.username=root
catalog.password=A627FC9A6DEA834C1FA777217871D09E
server.hss.bpmApplication=BPM\:29696
catalog.autoinit=true
server.datafile.dir=C:\\FlatFileSource
server.essbase.streamingCubeBuilding=true
server.timeoutPeriod=7200
logger.file=my_server.log
logger.limit=10000000
server.queueSize=220
server.threadCount=40
server.resourceCount=25
server.sql.fetchSize=100
server.tempDir=./Windows/Temp
server.readLockTimeOut=90
server.writeLockTimeOut=90
com.hyperion.cp.level=FINE

Configuring the Administration Services Console Launch File

A default EASLaunch.properties file is created during installation and contains the configuration setting necessary to launch Administration Services Console directly from deployed cubes in Essbase Studio Console.

After installation and configuration, you must edit EASLaunch.properties manually to add the remaining optional Essbase Administration Server properties.

Note:

You can configure only one Oracle Essbase Administration Services instance in EASLaunch.properties.
To configure the EASLaunch.properties file:

1 Locate and open the EASLaunch.properties file in the Essbase Studio installation in the \Console\eas directory.

   Note:
   Edit the file in text format with any text editor, such as Windows Notepad. Enter each setting on a separate line in the file. You do not need to end each line with a semicolon.

2 Note the setting already written in this file, EASHome.
   During the installation process, EASHome is populated with the location of your Administration Services installation on the client machine. Note that this is not the same as the location of the Administration Services client executable, but is equal to the value in the EASPATH environment variable. For example:
   EASHome=C:\Hyperion\products\Essbase\eas

   Note:
   This is the only required setting and assumes you are connecting to an instance of Essbase Administration Server on the local machine.
   Including only the EASHome setting in EASLaunch.properties means that, when launching Administration Services Console from a deployed cube in Essbase Studio Console, the Essbase Administration Services Logon window is displayed with the Essbase Administration Server field already populated with the server name of the local machine. To log in, you need only provide a valid user name and password.

3 Optional: If you want to connect to an instance of Essbase Administration Server on a different computer, add a line for EASServer and provide the machine name where Essbase Administration Server is installed.
   For example:
   EASServer=spruce1
   Including only the EASHome and EASServer settings in EASLaunch.properties means that, when launching Administration Services Console from a deployed cube in Essbase Studio Console, the Essbase Administration Services Logon window is displayed with the Essbase Administration Server field already populated with the server name you provided in this step. To log in, you need only provide a valid user name and password.

   Note:
   You can change the Essbase Administration Server setting (EASServer) at any time and relaunch Administration Services Console without stopping and restarting Essbase Studio Server.

4 Optional: Add lines for the user name and password settings.
   ● EASUser—A valid Administration Services user name. This setting can be overwritten at the console login screen with a valid user name. For example:
EASUser=admin

- EASPassword—The encrypted Administration Services password. This setting can be overwritten at the console login screen with a valid password. For example:

EASPassword=A627FC9A6DEA834C1FA777217871D09E

Including the optional user name and password setting in EASLaunch.properties means that, when launching Administration Services Console from a deployed cube in Essbase Studio Console, you bypass the Essbase Administration Services Logon window and go straight to Administration Services Console.

5 Save EASLaunch.properties.

Working with the Essbase Studio Server Startup File

The startServer.bat file contains the commands necessary to start Essbase Studio Server.

During the configuration process, the EPM System Configurator writes information to startServer.bat about the computer on which Essbase Studio installed so that Essbase Studio Server can start properly. Oracle recommends that you do not manually edit the startServer.bat file.

The startServer.bat is installed by default in:

HYPERION_HOME/products/Essbase/EssbaseStudio/Server

Example

An example of the startServer.bat file on a Windows 2003 operating system is shown below:

```bash
setlocal
set CONFIGTOOL_HOME=\%HYPERION_HOME\%common\config\9.5.0.0"
call "\%CONFIGTOOL_HOME\%setJavaRuntime.bat"
set PATH=\%HYPERION_HOME\%SAP\bin;\%PATH%
set ESSBASEPATH=C:\Hyperion\products\Essbase\EssbaseStudio\Server
%JAVA_HOME%\bin\java" -Xms128m -Xmx768m -DESS_ES_HOME="C:\Hyperion\products\ Essbase\EssbaseStudio\Server\ess_japihome" -jar "C:\Hyperion\ products\Essbase\EssbaseStudio\Server\server.jar"
```

Parameters

- set CONFIGTOOL_HOME
  
  Equals the path to the EPM System Configurator installation.

- call "\%CONFIGTOOL_HOME\%setJavaRuntime.bat"

  The path to the batch file used to call the Oracle's Hyperion Enterprise Performance Management System Configurator.

- set PATH=\%HYPERION_HOME\%SAP\bin;\%PATH%

  Contains a hard-coded path to the SAP drivers, and the PATH environment variable.

- "\%JAVA_HOME\%bin\java"

  The commands and parameters necessary start the Essbase Studio Server process.
Starting and Stopping Essbase Studio Server and Console

After you have set up the catalog database, and configured the server.properties and EASLaunch.properties files, you are ready to start the Essbase Studio Server and Console.

➤ To start Essbase Studio Server and Console:

1 On Windows, start the server from the Start menu by selecting Oracle EPM System, then Essbase, then Essbase Studio, then Start Server.

You can also start the server on Windows by navigating to <essbase_studio_home>\server\ and running the startServer.bat file.

2 Start the console from the Start menu by selecting Oracle EPM System, then Essbase, then Essbase Studio, then Essbase Studio Console.

On Windows, you can also start the console by navigating to <essbase_studio_home>\Console and running EssbaseStudio.exe.

3 In the Essbase Studio Login screen, enter a user name and password.

The default user name is admin; the default password is password.

4 To view the help, complete the procedure in “Configuring the Console to Locate Online Help Files” on page 40.

➤ To stop Essbase Studio Server and Console:

1 Stop the console by selecting the Essbase Studio Console File menu and selecting Exit.

2 On Windows, stop the server from the Start menu by selecting Oracle EPM System, then Essbase, then Essbase Studio, then Stop Server.

You can also stop the server on Windows by navigating to <essbase_studio_home>\server\ and running the stopServer.bat file.

Configuring the Console to Locate Online Help Files

Use this procedure to locate the Essbase Studio online help files if you have manually configured your system (not recommended) or if you have moved the help files to another location.

➤ To configure Essbase Studio Console to locate the online help:

1 In Essbase Studio Console, select Edit, then Preferences.

2 In Path to help files, click Browse and browse to the directory where you extracted the online help.

The help file location is used to direct the “Help Topics” command on the Help menu to the location of the online help. It is also used to the call the help when you press the F1 key in dialog boxes and console panes.
Tip:
Place the online help in a folder called “help” in a location easily accessible by Essbase Studio Console; for example, in the <essbase_studio>\help folder.

3 Click Apply, then click OK.

Essbase Studio Server Commands

Essbase Studio Server commands inform you about the tasks the server is performing and the state of the server in terms of the request manager, which controls the flow of tasks into Essbase Studio Server and the execution of the tasks. There are also commands that allow you to modify the task flow.

Essbase Studio Server commands are listed in the server window at startup. The commands are entered directly into the server window, along with any required response.

Some commands are used to make changes to the default server settings or to override the value you specified for the setting in server.properties. Changes you make to the settings using server commands are not persistent to the next Essbase Studio Server session. After restarting the server, any command settings you changed are reset to the default or to the value you specified for the setting in server.properties.

You can press Enter in the server window anytime for a list of available commands.

Essbase Studio Server commands are described below.

Server Commands

● version—Prints onscreen the Essbase Studio Server version information.

Command
version

Example
version

Returns
Oracle Essbase Studio Server version 11.1.1.0.00 Build Number 1206

● dumps—Displays stack traces of all server threads. This command is used mainly in the development environment.

Command
dumps

Example
dumps

Returns
The following is an example of a portion of the information returned when running the dumps command:
thread: Finalizer

    at java.lang.Object.wait(Native Method)
    at java.lang.ref.ReferenceQueue.remove(Unknown Source)
    at java.lang.ref.ReferenceQueue.remove(Unknown Source)
    at java.lang.ref.Finalizer$FinalizerThread.run(Unknown Source)

thread: Reference Handler

    at java.lang.Object.wait(Native Method)
    at java.lang.Object.wait(Unknown Source)
    at java.lang.ref.Reference$ReferenceHandler.run(Unknown Source)

thread: Signal Dispatcher

thread: Thread-1

    at java.lang.Object.wait(Native Method)
    at com.sun.jndi.ldap.pool.PoolCleaner.run(Unknown Source)

● pconf—Displays the state of the following Essbase Studio Server request manager parameters: the number of worker threads specified, the number of resources allocated, the number of resources available, the size of the queue, and the number of queues in use.

See “server.threadCount” on page 31 for a description of worker threads.

Command

pconf

Example

pconf

Returns

The following is an example of the information returned when running the pconf command.

Request manager configuration:
    Threads: 10, Resources: 5 (4 avail), Queue Size: 100 (10 in use)

● squeue—Sets the queue size in terms of number of tasks. The queue holds the tasks waiting to be executed by Essbase Studio Server.

The default number of tasks in the queue is 200.

Command

squeue

Example

squeue
    new queue size: 250

Note:

Any change you make to this setting is not persistent to the next Essbase Studio Server session. After restarting the server, the squeue parameter is either reset to the default or to the value you specified for server.queueSize, described in “Configuring the Server Properties File” on page 23.
● **sthd**—Sets the number of worker threads allocated to Essbase Studio Server.

See “server.threadCount” on page 31 for a description of worker threads.

**Note:**

To account for the worker threads that listen for and get tasks from Essbase Studio clients, the number of workers threads must be set to a number greater than the number of resources.

The default setting is 30 threads.

**Command**

sthd

When you run this command, you are prompted to enter the new thread count.

**Example**

sthd

`new thread count: 45`

**Note:**

Any change you make to this setting is not persistent to the next Essbase Studio Server session. After restarting the server, the `sthd` parameter is either reset to the default or to the value you specified for `server.threadCount`, described in “Configuring the Server Properties File” on page 23.

● **sres**—Sets the number of resources, which defines the number of tasks that can be executed concurrently.

The default setting is 20 resources.

**Command**

sres

When you run this command, you are prompted to enter the new number of resources.

**Example**

sres

`new resource count: 30`

**Note:**

Any change you make to this setting is not persistent to the next Essbase Studio Server session. After restarting the server, the `sres` parameter is either reset to the default or to the value you specified for `server.resourceCount`, described in “Configuring the Server Properties File” on page 23.

● **exit**—Stops Essbase Studio Server and closes the server window.

**Command**

exit
Example

exit
About Using Shared Services with Essbase Studio

Essbase Studio user management and security is provided through Shared Services, which provides user management, user provisioning, and external authentication definition. Provisioning refers to the process of assigning roles and access permission to Essbase Studio users.

Products that implement Shared Services functionality require access to a Shared Services server running Shared Services client and server software, and to database dedicated to Shared Services.

Essbase Studio Roles for Shared Services

Roles determine the tasks that users can perform. Roles can be grouped in the following ways:

- **Product-specific roles**
  
  Examples of Essbase Studio are cpAdmin and cpDM. The roles determine the type of interaction that the user can have with Essbase Studio artifacts. The interaction of each role with specific artifacts is described in the *Essbase Studio Provisioning* appendix in the Oracle Hyperion Enterprise Performance Management System Security Administration Guide.

- **Shared Services roles**
  
  Examples of Shared Services roles are Project Manager or Provisioning Manager. Most Shared Services roles are global (the role applies to all Shared Services applications). For information on Shared Services roles, see the *Oracle Hyperion Enterprise Performance Management System Security Administration Guide*.

The following Essbase Studio roles provide different levels of authority to perform tasks in Essbase Studio:

- **cpAdmin**—Administrator; performs all Essbase Studio tasks, including deploys cubes and executing drill-through reports.
● cpDM—Data Modeler; performs all tasks related to metadata element creation and maintenance; deploys cubes; executes drill-through reports

● cpDSAdmin—Data Source Administrator; performs all tasks related to data source connection creation and maintenance; executes drill-through reports

● cpViewer—Viewer; views all Essbase Studio data sources and metadata elements; executes drill-through reports

Because the cpAdmin and cpDM users are responsible for deploying cubes to Essbase, they must be provisioned with these additional roles:

● Shared Services Administrator (optional)
● Shared Services Project Manager (required)

At a minimum, the Project Manager role is required to deploy cubes.

● Essbase Administrator (optional)
● Essbase Create/Delete Application (required)

When deploying cubes, information is written into Essbase; therefore, at a minimum, the Create/Delete Applications role is required in order to write to Essbase.

After cubes are deployed, you are not required to provision Essbase Studio users for access to the new Essbase applications and databases. Permissions are inherited for the cubes Essbase Studio deploys to Essbase. You may, however, have to provision Essbase users for access to applications and databases created in Essbase Studio.

### Launching and Logging In to Shared Services Console

The procedure for launching Oracle's Hyperion® Shared Services Console is in the Essbase Studio Provisioning appendix in Oracle Hyperion Enterprise Performance Management System Security Administration Guide.

When you launch Shared Services Console, you log in as whichever user is appropriate. For example, you must log in as a Shared Services Administrator to provision Essbase Studio users.

### Assigning Access to Users in Shared Services

After installation and configuration, you assign Essbase Studio roles to users and groups in Shared Services Console.

To manage Essbase Studio users in Oracle's Hyperion® Shared Services Console, you must log in to the console as a user who is provisioned with the Shared Services Provisioning Manager role.

When provisioning users, Essbase Studio roles are listed under the Essbase Studio Server project.
Note:

Essbase Studio users or groups that will perform cube deployment must also be assigned, at a minimum, the role of Shared Services Project Manager.

Shared Services supports aggregated groups, in which a parent group contains one or more subgroups. The subgroups inherit the roles of their parent group. For example, if a parent group is provisioned with the Essbase Studio data modeler role, cpDM, any subgroups (and users in the groups) inherit the cpDM role.

To assign access to users and groups and view a report of provisioned roles for users and groups, see the Oracle Hyperion Enterprise Performance Management System Security Administration Guide.
Overview

In Essbase Studio, you connect to various types of data sources using the Connection Wizard. Data sources can be relational databases, Oracle Business Intelligence Enterprise Edition sources, Performance Management Architect cubes, or text file data sources. The Connection Wizard enables you to define a data source for modeling and optionally, to populate a minischema, and to create metadata elements in the catalog.

You can also use the Connection Wizard to set up connections to Essbase Server instances, which you can use later when you deploy cubes.

Setting up a data source in the Connection Wizard is the first task you perform after installing Essbase Studio. Before you begin creating metadata elements, hierarchies, cubes, and other application objects, you must define the source for their data. The first page of the Connection Wizard collects information from you about the data source (data source type, user name, password, server name, and other information), and then “scrapes” the data source to store the information about data source tables, columns and joins in the catalog. Subsequent steps in the wizard enable you to create and populate a minischema, create metadata elements in the catalog, and bind those metadata elements to the physical elements in the data source.
Once a data source connection is created, the connection is always present in Essbase Studio (until you delete it); you need not reconnect to the data source upon subsequent logins to the Essbase Studio Server and Console.

**Note:**

When creating or editing data source connections, you are asked for a database user name and password for the data source. You must provide a user name that has at least read permission to the data sources (databases) to which you are connecting.

To get started, see “Creating Connections with Connection Wizard” on page 50.

**Creating Connections with Connection Wizard**

Use the Connection Wizard to create connections to many types of data sources, including relational and text. When you use the Connection Wizard, you accomplish several tasks:

- Define the connection parameters for a data source.
  
  See “Defining Connection Parameters” on page 50.

- Optionally, select which source tables or files to include in the connection.
  
  See “Selecting Tables to Include in the Connection” on page 51.

- Select whether to create a minischema for this data source.
  
  See “Selecting a Minischema Option” on page 51.
  
  If you choose to create a minischema, you will also populate it.
  
  See “Populating a Minischema” on page 51.

- Create metadata elements from the data source elements.
  
  See “Creating Metadata Elements” on page 52.

These procedures provide the basis for creating customized metadata elements, such as dimension elements, and hierarchies, which you can use to create cube schemas and Essbase models, and perform cube deployments.

**Note:**

All steps after defining the data source parameters in the first page of the Connection Wizard are optional.

**Defining Connection Parameters**

You define connection parameters in the first page of the Connection Wizard. The parameters you define depend on the type of data source to which you want to connect.
Relational data sources—Includes relational sources such as Oracle, IBM DB2, Microsoft SQL Server and MySQL; also includes Oracle Business Intelligence, and Teradata. See “Defining Connection Parameters for Relational Sources” on page 53.

Essbase Server instances—See “Creating Connections to Essbase” on page 59.

Dimension Server sources—Used for Performance Management Architect sources; see “Creating Connections to Performance Management Architect Data Sources” on page 59.

Flat or text file data sources—See “Defining Connection Parameters for Text File Sources” on page 61.

Selecting Tables to Include in the Connection

When creating a data source connection to a relational data source, you can include all tables, views, aliases, and synonyms from the data source or including only a subset of the tables.

Relational data sources include relational sources such as Oracle, IBM DB2, Microsoft SQL Server, and MySQL; also includes Oracle Business Intelligence, and Teradata. See “Defining Connection Parameters for Relational Sources” on page 53.

Selecting a Minischema Option

You can create a minischema during the data connection process. Minischemas created from relational data sources include joins between tables. Minischemas created from Oracle Business Intelligence, Dimension Server, or text file sources do not include joins between tables, but the minischema created from these sources can be a useful visual tool.

- Relational data sources—Includes relational sources such as Oracle, IBM DB2, Microsoft SQL Server, and MySQL; also includes Oracle Business Intelligence and Teradata. See “Selecting a Minischema Option for Relational Sources” on page 55.
- Essbase Server sources—See “Creating Connections to Essbase” on page 59.
- Dimension Server sources—Includes Performance Management Architect sources; see “Creating Connections to Performance Management Architect Data Sources” on page 59.
- Flat or text file data sources—See “Selecting a Minischema Option for Text File Sources” on page 64.

Populating a Minischema

When you choose to create or reuse a minischema during the data source connection process, choose which source elements to use to populate your minischema. You may choose all elements or a subset of elements, depending on your business needs.
Creating Metadata Elements

During the data source creation process, you can create metadata elements from the physical elements in your data source. The metadata elements generally represent the tables and columns in your data source and are the basis to later create other metadata elements such as customized dimension elements and hierarchies.

See the following topics for information on creating metadata elements for a particular data types:

- Relational data sources—including relational sources such as Oracle, IBM DB2, Microsoft SQL Server and MySQL; also includes Oracle Business Intelligence and Teradata; see “Creating Metadata Elements for Relational Sources” on page 58
- Dimension Server sources—includes Performance Management Architect sources; see “Creating Metadata Elements for Performance Management Architect Sources” on page 60
- Flat or text file data sources—see “Creating Metadata Elements from Text File Sources” on page 66

Note:
This activity is not applicable to Essbase data sources.

After you create a connection, the connection information is always present in Essbase Studio; you need not reconnect to the data source on subsequent logins to the Essbase Studio Server and Console.

Creating Connections to Relational Sources

Use the procedures in this section to connect to a relational data source.

These are the relational data sources to which you can connect:

- Oracle
- IBM DB2
- Microsoft SQL Server
- Teradata
You may also connect to Oracle Business Intelligence sources as you would relational sources; however, you may not see all the corresponding physical tables in the source. You may see views instead.

*Note:* MySQL can be used as a data source if you have installed the MySQL JDBC connector.

These topics cover the workflow for creating a connection to a relational data source:
- “Defining Connection Parameters for Relational Sources” on page 53.
- “Selecting Tables for Relational Sources” on page 54.
- “Selecting a Minischema Option for Relational Sources” on page 55.
- “Populating a Minischema for Relational Sources” on page 57.
- “Creating Metadata Elements for Relational Sources” on page 58.

### Defining Connection Parameters for Relational Sources

To define the parameters of a data source:

1. In the **Source Navigator**, select **Data Sources**.
2. Right-click **Data Sources** in the physical tree, and select **New**, then **Data Sources**. The Define Parameters page of the Connection Wizard is displayed.
3. Enter a **Connection Name**.
4. Enter an optional **Description**.
5. Select the appropriate **Data Source Type**.
   - For example, if you are creating a connection to a Microsoft SQL Server data source, select Microsoft SQL Server from the drop-down list.
6. In **Server Name**, enter the name of server where the database resides.
7. **Oracle users only**—If you are connecting to an Oracle database, in the **Oracle SID/Service Name** group, enter the **SID** or **Service Name** for your Oracle instance as follows:
   - For Oracle 8 and below, enter the Oracle SID
   - For Oracle 8i (8.1) and above, enter the either the Oracle SID or the Oracle Service Name
8. **Optional**: To use a port number other than the default, clear the **Default** check box next to **Port** and enter the correct port number in the text box.
   - If you are using the default port number, you can skip this step.
9. Enter the **User Name** and **Password** for this database.
10. In **Database Name**, select the name of the database to which you want to connect.
If you do not know the name of the database to which you want to connect, click the “Fetch database” button next to the Database Name. Select the database from the list of the databases available on the server you designated in step 6.

**DB2 users only**—You must type the database name; you are not presented with a list of databases from which to choose.

11. **Click Test Connection.**

If the information you entered in the wizard is correct, a message confirms a successful connection.

If you entered incorrect information in the wizard, a message is displayed explaining that invalid credentials have been provided. Correct the errors and retest until the connection is successful.

12. **Click Next or Finish.**

Clicking Next takes you to the Select Tables page of the wizard, described in “Selecting Tables for Relational Sources” on page 54.

### Selecting Tables for Relational Sources

When you are creating a connection to a relational source data source, you can include all available tables or a subset of the tables in your data source connection. You can also choose to include all available views, alias tables, and synonyms, or a subset of those objects. Once your selections are made, Essbase Studio scrapes the data source for table, column and join information.

**Note:**

In this procedure, “tables” refers to tables, views, aliases, and synonyms.

To select tables for a relational data source connection:

1. In the Select Tables page of the Connection Wizard, click one or more of the buttons to the left of Available tables to control the content displayed in the Available tables box:
   - Show tables
   - Show views
   - Show aliases
   - Show synonyms

   These button are toggled. Click a button again to stop displaying a particular group of tables. For example, if you had clicked “Show views” to display database views, click “Show views” again to stop displaying them.

2. **Optional:** Enter a Filter to limit the tables displayed in Available tables, and click Apply.

   For example, if database tables are prefixed for a certain business area, such as “perf” for Performance, enter a filter to return only those tables in that area, such as: `perf*`
Filters apply to all tables, views, aliases, or synonyms you have chosen to display.

3 Perform an action:

- Select the tables you want to include in the data source connection and click the **Add selected tables to data source** button.
  
  The tables you selected should appear in the “Tables in Data Source” box.

- Click the **Add all tables to data source** button to add all tables to the Tables in Data Source box.
  
  All tables should appear in the “Tables in Data Source” box.

4 Optional: Remove tables from the **Tables in Data Source** box by performing one of the following tasks:

- Select the tables you want to exclude from the data source connection and click the **Remove selected tables from data source** button.
  
  The tables you selected should appear in the “Available Tables” box.

- Click the **Remove all tables from data source** button to remove all tables from the Tables in Data Source box.
  
  All tables should appear in the “Available Tables” box.

5 Click **Next** or **Finish**.

**Note:**

Clicking Next or Finish begins the data source scraping process. If you have selected a large number of tables to add to the connection, the data source scraping process can take several minutes.

Clicking Next takes you to the Select Minischema page of the wizard, described in “Selecting a Minischema Option for Relational Sources” on page 55.

**Note:**

If you later decide that you want to add tables to a data source that contains only a subset of tables, you may do so by performing an incremental update of the data source. See “Performing an Incremental Update of an Existing Data Source Connection” on page 67.

### Selecting a Minischema Option for Relational Sources

In the Select Minischema page of the Connection Wizard, you can perform the following actions:

- **Create a new minischema diagram**

  **Note:**

  If you are accessing a relational data source, you can also select the “Use Introspection to Detect Hierarchies” option when you choose to create a new minischema diagram. Selecting this option provides a deeper inspection of the data source to return more details, such as
Creating a Minischema for Relational Sources

You may create a minischema with or without the introspection option selected. This topic describes both procedures.

➤ To create a new minischema without introspection:

1. In Select Minischema, choose Create a new schema diagram.
2. Enter a name for the minischema, or accept the default name provided.
3. Click Next or Finish.

Clicking Next takes you to the Populate Minischema page of the wizard, as described in “Populating a Minischema for Relational Sources” on page 57.

Reusing an Existing Minischema for Relational Sources

➤ To reuse an existing minischema:

1. Select Reuse a schema diagram
2. From the drop-down list, select the schema diagram you want to reuse.
3. Optional: To clear the contents of the diagram you chose in step 2, select the Clear existing schema check box.
4. Click Next or Finish.

Clicking Next takes you to the Populate Minischema page of the wizard, as described in “Populating a Minischema for Relational Sources” on page 57.
Skipping Minischema Creation for Relational Sources

To skip creating a minischema:

1. Select **Skip minischema diagram**.
2. Click **Next** or **Finish**.

Clicking Next takes you to the Create Metadata Elements page of the wizard, as described in “Creating Metadata Elements for Relational Sources” on page 58.

Populating a Minischema for Relational Sources

Choose from the list of available tables in the data source to populate the minischema. You can apply filters and add related objects to the tables chosen for the minischema.

**Note:**

The tables you selected to include in the data source in “Selecting Tables for Relational Sources” on page 54 are included by default in the “Tables in schema” box in the Populate Minischema page of the Connection Wizard.

To populate the minischema:

1. In **Populate Minischema**, from the **Tables in schema** list, select the tables you DO NOT want to include in the minischema, then click the **Remove selected tables from the schema** button to move the tables to the **Available Tables** list.

To move all tables to the Available Tables list, click the “Remove all tables from the schema” button.

**Note:**

In the Select Minischema page of the wizard, if you chose to reuse a minischema and selected the Clear existing schema check box, you must perform this step (see step 3 in “Reusing an Existing Minischema for Relational Sources” on page 56).

2. To add tables to the minischema, from the **Available Tables** list, select the tables you to include, then click the **Add selected tables to the schema** button to move the tables to the **Available Tables** list.

3. **Optional:** To apply a filter to tables listed in **Available Tables**, enter the filter in the **Filter** text box, and click **Apply**.

**Note:**

Filters apply only to tables listed in the Available Tables list.

4. **Optional:** To add related objects to the minischema, select one or more tables in the **Tables in Schema** list, and click the **Add Related Objects** button.
For example, if you added a single table to the Tables in Schema list, selecting that table and clicking Add Related Objects adds to the list any tables that are joined to that table.

5 **Optional: To populate the minischema with tables from another data source:**
   a. In **Connections**, select the data source to which you want to connect.
   b. In **Databases**, select the database from which you want to select tables for this minischema.
   c. Repeat step 1 through step 4 for all data sources from which you want to add tables to this minischema.

6 **Click Next or Finish.**
   Clicking Next takes you to the Create Metadata Elements page of the wizard, as described in “Creating Metadata Elements for Relational Sources” on page 58.

### Creating Metadata Elements for Relational Sources

Create metadata elements from the tables and columns in your data source. These metadata elements can be used later to create dimension elements and hierarchies, which you can then use to build a cube schema.

➤ To create metadata elements:

1 **In Create Metadata Elements, in the Available Source Objects list, select which tables and columns to add to the folder for this data source in the Metadata Navigator.**
   - To create metadata elements for all columns within all tables, select the check box next to the data source name at the top of the Available Source Objects list.
   - To create metadata elements for selected tables, including all their columns, select the check box next to the table names that you want to add.
   - To create metadata elements for selected columns of a table, expand the table and select the check boxes next to the names of the columns that you want to add.

2 **Optional: To store the metadata elements in a folder other than the default folder:**
   a. Click **Browse**.
   b. In **Select Folder**, navigate to the folder in which you want to store the metadata elements, expanding the folders as necessary.
      Alternatively, click New Folder, enter the folder name in the text box, and click OK.

3 **Click Finish.**
   If you created a minischema, view the new minischema in the Minischema work area of the Essbase Studio Console.
   If you created metadata elements, view the new metadata elements listed in the Metadata Navigator.
Creating Connections to Essbase

You can create a connection to a specific Essbase Server instance, which can be used later during the cube deployment process.

➤ To define a connection to an Essbase instance:

1 In the Source Navigator, select the Data Sources tab.
2 Right-click Data Sources in the physical tree, and select New, then Data Source.
3 Enter a Connection Name.
4 Optional: Enter a Description.
5 In Data Source Type, select Essbase Server.
6 Under Parameters, in Essbase Server, enter the name of computer where this Essbase Server instance resides.
7 Optional: To use a port number other than the default, clear the Default check box next to Port and enter the correct port number in the text box.

   If you are using the default port number, skip this step.
8 Enter the User Name and Password for this Essbase Server instance.
9 Click Test Connection.
   If the information you entered in the wizard is correct, a message is displayed confirming a successful connection.
   If you entered incorrect information in the wizard, a message is displayed explaining that invalid credentials were provided. Correct the errors and retest until the connection is successful.
10 Click Finish.
   You should see the connection name in the Source Navigator. You can select this name as the Essbase Server Connection later, at cube deployment time.

Creating Connections to Performance Management Architect Data Sources

These topics discuss the workflow for creating connections to Performance Management Architect data sources:

- “Defining Connection Parameters for Performance Management Architect Sources” on page 60.
- “Creating Metadata Elements for Performance Management Architect Sources” on page 60.
Defining Connection Parameters for Performance Management Architect Sources

To define a connection to a Performance Management Architect data source:

1. In the **Source Navigator**, select the **Data Sources** tab.
2. Right-click **Data Sources** in the physical tree and select **New**, then **Data Source**.
3. Enter a **Connection Name**.
4. **Optional**: Enter a **Description**.
5. In **Data Source Type**, select **Dimension Server**.
   
   Performance Management Architect is an example of a dimension server.
6. Under **Parameters**, in **Server Name**, enter the name of computer where Performance Management Architect resides.
7. **Optional**: To use a port number other than the default, clear the **Default** check box next to **Port** and enter the correct port number in the text box.
   
   If you are using the default port number, skip this step.
8. Enter the **User Name** and **Password** for this Performance Management Architect instance.
9. **Click Test Connection**.
   
   If the information you entered in the wizard is correct, a message is displayed confirming a successful connection.
   
   If you entered incorrect information in the wizard, a message is displayed explaining that invalid credentials were provided. Correct the errors and retest until the connection is successful.
10. **Click Next or Finish**.

   Clicking Next takes you to the Create Metadata Elements page of the wizard, as described in “Creating Metadata Elements for Performance Management Architect Sources” on page 60.
   
   If you clicked Finish, you should be able to expand the data source name in the Source Navigator to view the dimensions and levels of this data source.

Creating Metadata Elements for Performance Management Architect Sources

Create metadata elements from the physical tables Performance Management Architect in your data source. These metadata elements can be used later to create dimension elements and hierarchies, which you can then use to build a cube schema.

To create metadata elements:

1. In **Create Metadata Elements**, in the **Available Source Objects** list, select which objects to add to the folder for this data source in the **Metadata Navigator**.
To add all objects to the Metadata Navigator, select the check box next to the data source name at the top of the Available Source Objects list.

To add only selected elements to the Metadata Navigator, select the check box next to the names of the elements that you want to add.

Alternatively, to add particular children of an object to the Metadata Navigator, expand the element and select the check boxes next to the names of the children objects that you want to add.

2 Optional: To store the metadata elements in a folder other than the default folder:
   a. Click Browse.
   b. In Select Folder, navigate to the folder in which you want to store the metadata elements, expanding the folders as necessary.
      Alternatively, click New Folder, enter the folder name in the text box, and click OK.

3 Click Finish.

You should be able to view the new metadata elements listed in the Metadata Navigator. Additionally, expand the data source name in the Source Navigator to view the dimensions and levels of this data source.

Creating Connections to Text File Data Sources

The process for creating text file data sources is slightly different from that of creating relational data sources. To create a data source connection to a flat file source, complete these procedures:

- “Defining Connection Parameters for Text File Sources” on page 61.
- “Modeling Text Files” on page 63.
- “Selecting a Minischema Option for Text File Sources” on page 64.
- “Creating Metadata Elements from Text File Sources” on page 66.

Defining Connection Parameters for Text File Sources

In this page of the wizard, you define the connection parameters of your text file data source.

Note:
Essbase Studio does not support Microsoft Excel files as data sources.

➤ To create a flat file data source:

1 In the Source Navigator, select the Data Sources tab.
2 Right-click Data Sources in the physical tree and select New, then Data Source.
3 Enter a Connection Name.
4 Optional: Enter a Description.
5 Select Text File as the Data Source Type.

6 To provide the Location of the text file data source, click Browse.

7 In the Text File Location dialog box, select the directory that contains the text file data source you want to access.

Note:
You can select only one directory per data source. Essbase Studio does not support text files from different subdirectories within the same directory.

Note:
Text file data source directory names must not contain spaces.

8 Review the files listed under Contents of selected directory to ensure that this is the text file data source you want to select.

9 When you have made your selection in the Text File Location dialog box, click OK to return to the Define Connection Parameters page of the Connection Wizard.

10 Optional: In Skip records, enter the number of records to skip from the beginning of each text file in the data source.

   For example, the text files in your data source may contain identifying comments as the first three lines of each file. In this case, enter 3 in Skip Records.

   The selection you make here can be overridden on a file-by-file basis in the Model Text Files page of the Connection Wizard.

11 Optional: Select Column names in first row if the first row after any skipped records contains column names.

   If you do not select this check box, the Connection Wizard assigns a default name to each column from each file. You can change the default naming later on a file-by-file basis when you model the text files, as described in “Modeling Text Files” on page 63.

12 Select a Delimiter:
   ● Comma
   ● Tab
   ● Space
   ● Custom—Specify a delimiter if your data source uses a delimiter other than a comma, tab or space.

   The delimiter you specify is used as the default for future text file connections. However, you can change this setting for specific text file data sources in the Model Text Files page of the wizard.

13 Click Next to proceed to the Model Text Files page of the wizard (described in “Modeling Text Files” on page 63).
Modeling Text Files

In this page of the wizard, you select which text files you want to model for the data source. You can view records in each file individually and perform modeling tasks on a file-by-file basis.

➢ To model the text files:

1 Select the check box next to a text file name, then highlight the name to view a sample of the records in the lower text box.

Essbase Studio displays up to 19 records in the lower text box.

You can also click the Select All button to select all files in the data source, then highlight the name of each file to view the records.

Note:

To clear all check marks, click Clear All. You can also click a selected check box to clear it.

2 Optional: In the Filter text box, enter a filter for the text file set.

For example, to filter for text files prefixed with “prod,” enter prod*.txt. A list of text files beginning with “prod” is displayed.

3 Optional: Perform modeling as necessary on the appropriate files, selecting the check box next to the text file and highlighting the file name.

a. In Skip records, enter the number of records to skip from the beginning of each text file in the data source.

   For example, the text files in your data source may contain identifying comments as the first three records of each file. In this case, enter 3 in Skip Records.

   Note:

   There is a 1000 row limit on the number records that can be skipped.

b. Select Column names in first row if the first row after any skipped records contains column names.

   If you do not select this check box, the Connection Wizard assigns a default name to each column from each file.

c. Select a Delimiter:

   ● Comma
   ● Tab
   ● Space
   ● Custom—Specify a delimiter if your data source uses a delimiter other than comma, tab, or space.

4 Optional: To override the column data type for a column, click its header in the work area to view the context menu where you select a new data type.

Available data types are:
● Text
● Integer
● Large integer
● Decimal

5 **Optional:** To hide a column, click its header in the work area to view the context menu and select **Hide column**.

To show previously hidden columns in the work area, select “Show hidden columns” from the context menu.

6 **Optional:** To change the column name and data type on multiple columns of a selected file:
   a. Click the first column header and select **Properties** from the context menu to launch the **Column Properties** dialog box.
   b. To change the column name, enter a new name in the **Name** text box.
   c. From the **Data Type** group, select a column type for the column selected.

   **Note:**
   Changing a data type from Text to Integer results in an error when viewing sample data; however, cube deployment completes successfully.
   d. Click **Next** to display the properties of the next column in the file.
   e. Repeat step 6.b through step 6.d until all necessary column data types have been changed, then click **OK** to return to the **Model Text Files** dialog box.

7 Click **Next** to view the **Select Minischema** page of the **Connection Wizard**, as described in “Selecting a Minischema Option for Text File Sources” on page 64.

**Selecting a Minischema Option for Text File Sources**

You can create a minischema for a text file data source, and only self-joins within a file are supported. However, a minischema for a text file data source can still be useful for viewing files.

In the Select Minischema page of the Connection Wizard, you can perform the following actions:

● **Create a new minischema diagram**; the minischema will not contain joins between files.
  See “Selecting a Minischema Option for Text File Sources” on page 65,
● **Reuse an existing minischema diagram**, with an option to clear the existing schema,
  See “Reusing an Existing Minischema for Text File Sources” on page 65,
● **Skip creation of a minischema diagram**,
  See “Skipping Minischema Creation for Text File Sources” on page 65,

**Note:**
The “Use Introspection to detect hierarchies” option is not supported for text files.
Selecting a Minischema Option for Text File Sources

➤ To create a minischema:

1 In **Select Minischema**, choose **Create a new schema diagram**.
2 Enter a name for the minischema, or accept the default name provided.
3 Click **Next** or **Finish**.

Clicking Next takes you to the Populate Minischema page of the wizard, described in “Populating a Minischema for Text File Sources” on page 65.

Reusing an Existing Minischema for Text File Sources

➤ To reuse a minischema:

1 In **Select Minischema**, select **Reuse an existing schema diagram**.
2 From the drop-down list, select the schema diagram that you want to reuse.
3 Optional: To clear the contents of the diagram that you chose in step 3, select **Clear existing schema**.
4 Click **Next** or **Finish**.

Clicking Next takes you to the Populate Minischema page of the wizard, described in “Populating a Minischema for Text File Sources” on page 65.

Skipping Minischema Creation for Text File Sources

➤ To skip the minischema creation process, in **Select Minischema**, ensure that the **Skip Minischema Diagram** option is selected, then click **Next**.

The Create Metadata Elements page of the Connection Wizard is displayed, as described in “Creating Metadata Elements from Text File Sources” on page 66

Populating a Minischema for Text File Sources

If you created a minischema for your text file data source, choose from the list of available physical tables (files) in the data source to populate the minischema. You can also apply filters and add related objects to the files chosen for the minischema.

➤ To populate the minischema:

1 In **Populate Minischema**, from the **Tables in schema** list, select the tables you DO NOT want to include in the minischema, then click the **Remove selected tables from the schema** button to move the tables to the **Available Tables** list.

To move all tables to the Available Tables list, click the “Remove all tables from the schema” button.
In the Select Minischema page of the wizard, if you reused a minischema and selected the Clear existing schema check box, you must perform this step (see step 3 in “Reusing an Existing Minischema for Relational Sources” on page 56).

2 To add tables to the minischema, from the Available Tables list, select the tables you want to include, then click the Add selected tables to the schema button, to move the tables to the Available Tables list.

3 Optional: To apply a filter to tables listed in Available Tables, enter the filter in the Filter text box, and click Apply.

Note:
Filters apply only to tables listed in the Available Tables list.

Note:
The Add Related Objects button does not apply to files in a text file data source.

4 Optional: To populate the minischema with objects from another data source:
   a. In Connections, select the data source to which you want to connect.
   b. In Databases, select the database from which to select objects for this minischema.
   c. Repeat step 1 through step 4 for all data sources from which you want to add objects to this minischema.

5 Click Next or Finish.
   Clicking Next takes you to the Create Metadata Elements page of the wizard, described in “Creating Metadata Elements from Text File Sources” on page 66.

Creating Metadata Elements from Text File Sources

Create metadata elements from the contents of the files in your text file data source. These metadata elements can be used later to create dimension elements and hierarchies, which you can then use to build a cube schema.

To create dimension elements from the physical objects of a text file data source:

1 In Create Metadata Elements, select elements to add to the folder for this data source in the Metadata Navigator.
   ● To create metadata elements for all columns within all tables, select the check box next to the data source name at the top of the Available Source Objects list.
   ● To create metadata elements for selected tables, including all their columns, select the check box next to the table names that you want to add.
   ● To create metadata elements for selected columns of a table, expand the table and select the check boxes next to the names of the columns that you want to add.

2 Optional: To store the dimension elements in a folder other than the default folder:
a. Click **Browse**.

b. In **Select Folder**, choose a folder from the **Metadata Navigator** in which to store the dimension elements, expanding the folders as necessary.

   Alternatively, click **New Folder**, enter the folder name in the text box, and click **OK**.

3 **Click Finish.**

If you created metadata elements, view them in the Metadata Navigator.

You are ready to begin creating other metadata elements, such as dimension elements and hierarchies.

---

**Working with Data Source Connections**

Perform these tasks from the Data Sources tab:

- “Performing an Incremental Update of an Existing Data Source Connection” on page 67.
- “Performing Introspection on an Existing Data Source Connection” on page 70.
- “Editing Data Source Connection Properties” on page 71.
- “Refreshing the Connections List” on page 76.
- “Deleting Connections” on page 76.

---

**Performing an Incremental Update of an Existing Data Source Connection**

If you set up a data source connection containing a subset of data source tables, you may determine later that you need to add additional data source tables to the connection. You can perform an incremental update of the data source to add the needed tables to the connection.

You can perform an incremental update to relational or text file data sources, described in the following topics:

- “Performing an Incremental Update of a Relational Data Source” on page 67
- “Performing an Incremental Update of a Text File Data Source” on page 69

---

**Performing an Incremental Update of a Relational Data Source**

This procedure only applies to relational data sources. If you are working with a text file data source, see “Performing an Incremental Update of a Text File Data Source” on page 69.

**Note:**

In this procedure, “tables” refers to tables, views, aliases, and synonyms.
To perform an incremental update of a relational data source:

1. In the **Source Navigator**, select the **Data Sources** tab.

2. Under **Data Sources** in the physical tree, right-click the relational data source to which you want to add tables, and select **Incremental Update**.

   The Incremental Update dialog box is displayed.

3. Click one or more of the buttons to the left of **Available tables** to control the content displayed in the **Available tables** box:
   - Show tables
   - Show views
   - Show aliases
   - Show synonyms

   These buttons are toggled. Click a button again to stop displaying a particular group of tables. For example, if you had clicked “Show views” to display database views, click “Show views” again to stop displaying them.

4. In the **Available tables** box, perform an action:
   - Select the tables you want to include in the data source connection and click **Add selected tables to data source**.
     
     The tables you selected appear in the “Tables in Data Source” box.
   - Click **Add all tables to data source** to add all tables to the **Tables in Data Source** box.
     
     All tables should appear in the “Tables in Data Source” box.

   **Note:**

   When you select a large number of tables to add to the connection, the incremental update process can take several minutes.

5. **Optional:** To remove tables that you have just added to the data source, in the **Tables in Data Source** box, perform an action:
   - Select the tables you want to exclude from the data source connection and click **Remove selected tables from data source**.
     
     The tables you selected for removal appear in the “Available tables” box.
   - Click **Remove all tables from the data source** to exclude all tables from the data source
     
     All the tables you have just added are now removed from the data source connection.

   **Note:**

   You can only remove tables that you added during the current incremental update session. You cannot remove tables that are already included in the data source. Tables that grayed out in the “Tables in Data Source” box are not eligible for removal.

6. **Optional:** Enter a **Filter** to view only a portion of the available data source tables.
For example, if database tables are prefixed for a certain business area, such as with “perf” for Performance, enter a filter to return only those tables in that area, such as:

perf*

7 When you are finished adding tables to the data source, click OK.

**Performing an Incremental Update of a Text File Data Source**

This procedure only applies to text file data sources. If you are working with a relational data source, see “Performing an Incremental Update of a Relational Data Source” on page 67.

**Note:**

This procedure only applies to text file data sources.

To perform an incremental update of a text file data source:

1 In the Source Navigator, select the Data Sources tab.

2 Under Data Sources in the physical tree, right-click the text file data source to which you want to add tables, and select Incremental Update.

   The Incremental Update dialog box is displayed.

3 Optional: In the Filter text box, enter a filter for the text file set.

   For example, to filter for text files prefixed with “prod,” enter prod*.txt. A list of text files beginning with “prod” is displayed.

4 In the list of text files, select the check box next to the text files that you want to add to the data source.

5 Follow these steps to perform modeling as necessary on the selected files:

   a. Highlight the file name of one of the selected text files in the list

   b. In Skip records, enter the number of records to skip from the beginning of each text file in the data source.

      For example, the text files in your data source may contain identifying comments as the first three records of each file. In this case, enter 3 in Skip Records.

   **Note:**

   There is a 1000 row limit on the number records that can be skipped.

   c. Select Column names in first row if the first row after any skipped records contains column names.

      If you do not select this check box, the Connection Wizard assigns a default name to each column from each file.

6 Select a Delimiter:

- Comma
- Tab
● Space
● Custom—Specify a delimiter if your data source uses a delimiter other than comma, tab, or space.

7 Optional: To override the column data type for a column, click its header in the work area to view the context menu where you select a new data type.

Available data types are:
● Text
● Integer
● Large integer
● Decimal

8 Optional: To hide a column, click its header in the work area to view the context menu and select Hide column.

To show previously hidden columns in the work area, select “Show hidden columns” from the context menu.

9 Optional: To change the column name and data type on multiple columns of a selected file:
   a. Click the first column header and select Properties from the context menu to launch the Column Properties dialog box.
   b. To change the column name, enter a new name in the Name text box.
   c. From the Data Type group, select a column type for the column selected.

   Note:
   Changing a data type from Text to Integer results in an error when viewing sample data; however, cube deployment completes successfully.
   d. Click Next to display the properties of the next column in the file.
   e. Repeat step 6.b through step 6.d until all necessary column data types have been changed, then click OK to return to the Incremental Update dialog box.

10 When you have selected and modeled all text files you are adding to the data source, click OK to close the Incremental Update dialog box.

Performing Introspection on an Existing Data Source Connection

Introspection is a method of inspecting a physical data source for metadata elements. When you perform introspection, structural information that already exists in the data source is inspected to detect fact tables, dimension tables, hierarchies, aliases, and attributes. The metadata elements derived from introspection are then used to create cube schemas and, optionally, Essbase models.

Introspection can be performed during the data source connection creation process. Introspection can also be performed on an existing data source connection.
For instructions on performing introspection, see Chapter 6, “Introspection.”

Editing Data Source Connection Properties

You can edit the data source connection information.

Note:
All steps and substeps in the following procedure are optional. The tasks you complete depend on which information you want to modify.

➤ To edit the properties of a data source:
1. From the Source Navigator, select the Data Sources tab.
2. Under Data Sources in the physical tree, right-click the data source to edit and select Properties.
3. To change the data source connection name, enter a new name in Connection Name.
4. To change the data source description, enter a new description in Connection Description.
5. Complete the following tasks in the Parameters section of the dialog box as appropriate for the data source type. Remember, these tasks are optional.

Relational sources:

Note:
You cannot change the database name. If you change the server connection information, the new server must host a database of the same type, name, and table structure as the original database.

a. In Server Name, modify the name of server where the database resides.

b. Oracle users only—If you are connecting to an Oracle database, in the Oracle SID/Service Name group, enter the SID or Service Name for your Oracle instance as follows:
   ● For Oracle 8 and below, enter the Oracle SID
   ● For Oracle 8i (8.1) and above, enter the either the Oracle SID or the Oracle Service Name

c. To modify the port number, ensure that the Default check box next to Port is cleared, and enter the new port number.

d. Modify the User Name and Password for this database.

e. Click Test Connection.

   If the information you entered in the dialog box is correct, a message is displayed confirming a successful connection.
If you entered incorrect information in the dialog box, a message is displayed explaining that invalid credentials were provided. Correct the errors and retest until the connection is successful.

**Essbase Server connections:**

a. In **Essbase Server**, modify the name of computer where the Essbase Server resides.

b. To modify the port number, ensure that the **Default** check box next to **Port** is cleared, and enter the new port number.

c. Modify the **User Name** and **Password** for this instance of Essbase Server.

d. Click **Test Connection**.

   If the information you entered in the dialog box is correct, a message is displayed confirming a successful connection.

   If you entered incorrect information in the dialog box, a message is displayed explaining that invalid credentials were provided. Correct the errors and retest until the connection is successful.

**Dimension Server sources:**

a. In **Server Name**, modify the name of computer where Performance Management Architect resides.

b. To modify the port number, ensure that the **Default** check box next to **Port** is cleared, and enter the new port number.

c. Modify the **User Name** and **Password** for this instance of Performance Management Architect.

**Text file sources:**

a. To modify the **Location** of the text file data source, click **Browse**.

b. In the **Text File Location** dialog box, select the directory that contains the text file data source you want to access.

   **Note:**

   You can select only one directory per data source. Essbase Studio does not support text files from different subdirectories within the same directory.

c. Review the files listed under **Contents of selected directory** to ensure that this is the text file data source you want to select.

6 **Click Apply**, and then click **OK**.

7 **If you changed the data source name and want to view the updated data source name in the Source Navigator**, right-click the data source connection name you just changed and select **Refresh**.

   The new name is displayed in the physical tree.
Viewing Properties of Source Tables and Columns

You can view the table and column properties of minischema tables, relational and text file data sources, and user-defined tables. See the following topics for more information:

- “Viewing Properties of Minischema Tables, Relational Source Tables and User-Defined Tables” on page 73.
- “Viewing Properties of Text File Source Files” on page 74.

Viewing Properties of Minischema Tables, Relational Source Tables and User-Defined Tables

➤ To view the properties of a minischema table, relational source table, or user-defined table:

1. From the Source Navigator, select the Minischema tab or the Data Sources tab.
2. Under Minischemas or Data Sources in the physical tree, navigate to the minischema table, relational source table, or user-defined table whose properties you want to view.
3. Right-click the table and select Properties to view the General properties.

These properties are displayed:

- Physical source table name
- Data source connection name
- Primary keys, if any
- Comment

To view the column properties of the table, click the Column tab. See “Viewing Properties of Minischema Columns, Relational Source Columns, and User-Defined Table Columns” on page 73. Also see “Viewing the Statement on Which a User-Defined Table is Based” on page 74.

4. Optional: If you are viewing properties of a minischema table from a minischema diagram, enter a Comment.

Note:

The Comment field is not available when viewing properties of relational tables or user-defined tables.

Viewing Properties of Minischema Columns, Relational Source Columns, and User-Defined Table Columns

➤ To view the properties of a relational source column or a column in a user-defined table:

1. From the Source Navigator, select the Minischema tab or Data Sources tab.
2. Under Minischemas or Data Sources in the physical tree, navigate to the minischema column, relational source column, or the column in a user-defined table whose properties you want to view.
3 Right-click the table and select Properties, and then select the Columns tab.

These column properties are displayed:

- **Name**—Column name
- **Table Name**—Only displayed when accessing the Properties dialog directly from a data source or minischema column.
- **Type**—Column type
- **Allow Nulls**—Whether nulls are allowed

**Note:**

If you are viewing the column properties from the Column tab of the Table Properties dialog box, a key symbol is displayed next to the key column, if any, for the table.

**Viewing the Statement on Which a User-Defined Table is Based**

To view the SQL statement on which a user-defined table is based:

1. From the Source Navigator, select the Data Sources tab.
2. Under Data Sources in the physical tree, navigate to the user-defined table whose SQL statement you want to view.
3. Right-click the table and select Properties, and then select the User-Defined Statement tab.

   The SQL statement on which the user-defined table is based is displayed.

**Viewing Properties of Text File Source Files**

To view the properties of a source text file:

1. From the Source Navigator, select the Data Sources tab.
2. Under Data Sources in the physical tree, navigate to the text file.
3. Right-click the file name and select Properties to view the General properties.

   To view the column properties of the file, see “Viewing Properties of Text File Source Columns” on page 75.

   These properties are displayed:
   
   - Physical source file name
   - Data source connection name
   - The column names in the file and their column data types
Viewing Properties of Text File Source Columns

To view the properties of a column in a text file source:
1 From the Source Navigator, select the Data Sources tab.
2 Under Data Sources in the physical tree, navigate to the text file source column.
3 Right-click the column name and select Properties, and then select the Columns tab.

These column properties are displayed:
- Physical column name
- Data source connection name
- Text file name
- Column data type

Viewing Sample Data

From the Data Sources tab of the Source Navigator, you can view the available sample data for up to 200 rows of a table in a data source or from individual source columns.

In the Metadata Navigator, you can also view up to 200 rows of the available sample data related to a dimension element.

To view sample data in a data source:
1 From the Source Navigator, select the Data Sources tab and expand the data source for which you want to view sample data to display the tables in the source.
2 Perform either or both of the following actions:
   - To view data for a table in the data source, right-click the table and select View Sample Data.
   - To view data for one column from a table in a data source, expand the table, right-click the appropriate column, and select View Sample Data.

   Alternatively, to view data for one or more columns within a table, press and hold the Ctrl key, click on the columns whose data you want to view, then right-click and select "View Sample Data."

The sample data you requested is displayed in a new tab in the work area of the console.

Note:
A maximum of 200 rows of sample data is returned.

3 To close the sample data window, click the X on the window tab.

To view sample data for dimension elements:
1 In the Metadata Navigator, navigate to the dimension elements for which you want to view sample data.
2 Right-click the metadata element and select **View Sample Data**.

The sample data you requested is displayed in a new tab in the work area of the console.

**Note:**

A maximum of 200 rows of sample data is returned.

3 To close the sample data window, click the X in the tab for this window.

### Refreshing the Connections List

To refresh the list of data source connections in the **Data Sources** tab of the **Source Navigator**, right-click the top item in the physical tree, “Data Sources,” and select **Refresh**.

### Deleting Connections

You can delete a data source connection only if there are no metadata elements created from the connection.

To delete a data source connection, in the **Source Navigator**, select the **Data Sources** tab, right-click the name of the data source you want to delete, and select **Delete**.

### Showing Friendly Names

**Note:**

This functionality is for Microsoft SQL Server data source connections only.

By default, when you connect to a Microsoft SQL Server data source, the database table names are displayed in the **Data Sources** tab of the **Source Navigator** in the following “friendly” format:

\[schema\_name.database\_name.table\_name\]

You can disable the Show Friendly Names option and display full table names in the following format:

\[database\_name.table\_name\]

To show the full names of tables in a SQL Server data source connection, from the **Data Sources** tab, right-click the data source connection name and select **Show Friendly Names**. The check mark is cleared.
To show the friendly names of tables in a SQL Server data source connection, from the Data Sources tab, right-click the data source connection name and select Show Friendly Names. A check mark shows that Show Friendly Names is selected.

Creating User-Defined Tables

Use this dialog box to define one or more user-defined tables in order to create a virtual “view” of the data from your database.

See “User-Defined Tables Overview” on page 77 and “User-Defined Table Examples” on page 78.

To access the User-Defined Table Dialog Box:

1. Highlight the name of the appropriate data source in the Source Navigator.
2. Select New, then User Defined Table.
3. Define the new user-defined table:
   - Connection—Select the data source to which the table will be added.
   - Table name—Enter a name for the table.
   - Table definition—Enter the SQL statement to create the user-defined table using the SQL syntax required by the data source. This statement defines the names of the tables in the virtual view and can include any number of source column names. It becomes the basis for a logical table in the data source.

   Note:
   Oracle recommends including the full path when specifying database tables.

User-Defined Tables Overview

User-defined tables are logical tables that you create in Essbase Studio, 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 models without altering your relational schema or modifying the SQL generated by Essbase Studio. If your database schema is not properly structured for use with Essbase Studio, user-defined tables enable you to build models and outlines without changing the database schema.

Essbase Studio verifies the SQL commands you use to create a user-defined table and enables you to establish a virtual view of the database schema.

Note:
You cannot create user-defined tables for text file data sources.
Note:
Oracle recommends including the full path when specifying database tables in the user-defined table SQL statement.

User-Defined Table Examples

In Essbase Studio, you create user-defined tables, which are similar to an RDBMS view, by entering SQL syntax in the Table Definition text box of the User-Defined Table Definition dialog box. The examples in this topic show the difference between the SQL statement you use in an RDBMS and the SQL statement you use in Essbase Studio.

The following example illustrates a SQL statement to create a view in an RDBMS:

```sql
CREATE VIEW
View_Prod_Proddim
(Caffeinated, FamilyID, Ounces, PackageType, ProductID, SKU, Family) as
SELECT
  a.caffeinated, a.familyid, a.ounces, a.pkgtype,
  a.productid, a.sku, b.family
FROM tbc.product a, tbc.productdim b
WHERE a.sku=b.sku
```

Using the example above, in Essbase Studio, when you create a user-defined table, the SQL syntax does not include the CREATE VIEW portion of the SQL statement. Your SQL table definition syntax begins with the portion of the statement starting from SELECT.

Note:
Oracle recommends including the full path when specifying database tables.

The following user-defined table SQL example is written for a database where table names are expressed in the format catalog.table:

```sql
SELECT
  a.caffeinated,
  a.familyid, a.ounces, a.pkgtype,
  a.productid, a.sku, b.family
WHERE a.sku=b.sku
FROM tbc.product a, tbc.productdim b
```

The following is the same example written for a database where table names are expressed in the format catalog.schema.table:

```sql
SELECT
  a.caffeinated,
  a.familyid, a.ounces, a.pkgtype,
  a.productid, a.sku, b.family
WHERE a.sku=b.sku
FROM tbc.user1.product a, tbc.user1.productdim b
```
Minischemas are graphical models of the tables or text files in one or more data source connections. You create a minischema during the data source connection creation process, or you can create a minischema later, from data sources to which you have already connected.

For most data source types, minischemas are displayed similar to a logical model in an entity relationship diagram, showing joins between tables and categorizing values in the source database as dimensions. You have the option of choosing the tables to include in a minischema, so the tables displayed can be a subset of the data from one or more data sources.

Minischemas are a way to bring in a subset of tables from one or more data source connections in order to create a subject area on which to base further modeling.

Note:
For text file sources, join relationships are not displayed in minischemas.

Tasks you can perform in the minischema are:
- Add or delete joins between tables, either manually or by using inspection, where you select from a list of possible joins
- View sample data from tables or individual columns
- Create metadata elements
- Add color to the table headings in the minischema diagram, which can be useful as a visual aid

Minischemas are discussed as part of the data source creation process in “Selecting a Minischema Option for Relational Sources” on page 55.

Figure 1 shows an example of a minischema.
You can create a minischema when creating a data source connection, as a step in the Connection Wizard. But you can also create minischemas apart from the Connection Wizard process.

You can create any number of minischemas per data source. Multiple minischemas are a convenient way to create alternate views of a specific data source. An alternate minischema may contain, for example, a subset of the objects in the data source or additional joins.

**Creating or Editing Minischemas**

To create or edit a minischema, you complete the following tasks:

- Name the new minischema and provide or edit a comment, as described in “Setting General Properties for Minischemas” on page 80.
- Set a connection and add tables to or remove tables from the minischema, as described in “Adding or Removing Tables in a Minischema” on page 81.
To set the general minischema properties:

1. From the Source Navigator, select the Minischema tab, and then perform an action:
   - Right-click Minischema in the physical tree structure and select New, then Minischema to launch the Minischema Wizard.
   - Under Minischema in the physical tree structure, right-click the minischema to edit and select Minischema Properties.

2. For new minischemas, enter a Minischema Name.
   If you are modifying a minischema, the Minischema Name is not editable.

3. Optional: Enter or edit the Description.

4. Perform an action:
   - If you are creating a new minischema, click Next.
     The Add/Remove Tables page of the Minischema Wizard is displayed, as described in “Adding or Removing Tables in a Minischema” on page 81.
   - If you are editing a minischema, click the Add/Remove Tables tab and follow the instructions in “Adding or Removing Tables in a Minischema” on page 81.
     Alternatively, to add tables in the graphical view of an existing minischema using a drag-and-drop method, see “Adding Tables in a Minischema” on page 86.

Adding or Removing Tables in a Minischema

After naming the minischema, as described in “Setting General Properties for Minischemas” on page 80, complete the tasks in this topic.

Note:
You may also access the Add/Remove dialog box from an existing minischema by right-clicking in the minischema work area and selecting “Add/Removing Tables.”

To add or remove tables from a minischema diagram:

1. In Connections in the Add/Remove Tables page, select the data source from which you want to create the minischema.
   The Databases field displays the name of the database associated with this data source connection. This field is not editable.
   The Available Tables list is populated with the tables from the selected data source.

2. Select the tables you want to include in the minischema and click the Add selected tables to the schema button to move them to the Tables in Schema list.
   To move all available tables to the Tables in Schema list, click the “Add all tables to the schema” button.
3 Optional: If you want to apply a filter to an object before moving it to the Tables in Schema list, enter the filter in the Filter text box.

4 Optional: To add related tables to the minischema, select one or more tables in the Tables in Schema list, and click the Add Related Tables button.

You may want to add relational tables if, when you first created the data source connection, you did not select all available tables during the schema population procedure (see “Populating a Minischema” on page 51).

5 Optional: To add to the minischema objects from another data source:
   a. In Connections, select an alternate data source.
   b. Repeat step 2 through step 4 for all data sources from which you want to add objects to this minischema.

6 To remove tables from the minischema, select the tables you want to remove from the Tables in Schema list and click the Remove selected tables from the schema button to move them to the Available Tables list.

   To move all tables in the schema to the Available Tables list, click the “Remove all tables from the schema” button.

7 Click Finish.

8 In the Minischema work area of the Essbase Studio Console, review the new minischema diagram.

9 Optional—To allow Essbase Studio to auto-arrange the tables in the minischema, click the Lay out schema button in the toolbar.

Note:
You may also auto-arrange the tables by right-clicking in the minischema work area.

Working with Minischemas

After you have created a minischema, the following tasks can be performed:

● “Saving a Minischema” on page 83
● “Opening a Minischema” on page 83
● “Removing Elements from a Minischema” on page 83
● “Adding or Editing Joins in a Minischema” on page 83
● “Adding Tables in a Minischema” on page 86
● “Viewing Sample Data” on page 87
● “Applying Color to Minischema Tables” on page 88
● “Creating Metadata Elements from Minischema Objects” on page 88
● “Viewing Minischemas” on page 89
● “Editing Properties of a Minischema” on page 93
● “Refreshing the Minischemas List” on page 94
● “Deleting Minischemas” on page 94

**Saving a Minischema**

➤ To save a minischema, right-click in the minischema work area and select **Save Minischema**.

**Opening a Minischema**

➤ To open an existing minischema for editing, in the **Minischema** tab of the **Source Navigator**, right-click the name of the minischema you are going to work with and select **Edit**.

**Removing Elements from a Minischema**

You can modify a minischema by removing tables from it.

You can follow the dialog box-based procedure described in “Adding or Removing Tables in a Minischema” on page 81 to specifically remove table objects from the minischema. You can also use the process described below to remove tables or selected joins.

➤ To remove tables or joins from a minischema:

1. If not already opened, open the minischema (see “Opening a Minischema” on page 83).
2. **Optional**—In the minischema work area, right-click the table object you want to remove and select **Remove**.
   
   Repeat this step for all tables you want to remove.
3. **Optional**—To remove a join, right-click the join in the minischema work area and select **Remove**.
   
   Repeat this step for all joins you want to remove.
4. To save the minischema, right-click in the minischema work area and select **Save Minischema**.

**Adding or Editing Joins in a Minischema**

You can add joins to minischema tables manually or you can have Essbase Studio perform an inspection of your data source to determine possible logical joins for you.

● To add joins manually, see “Adding or Editing Joins” on page 84
● To add joins by inspection, see “Adding Joins By Inspection” on page 86
Adding or Editing Joins

Essbase Studio uses the same dialog box to manually create and edit joins. Once created, you can edit joins by changing the columns used in a join pair, adding more join pairs, or modifying other criteria. See “Adding or Editing Joins” on page 84.

In the Edit Properties of Minischema Join dialog box, you can perform the following join operations:

- Add or edit joins between relational tables
- Add or edit self joins within a single relational table or flat file

To add or edit a join:

1. If not already opened, open the minischema (see “Opening a Minischema” on page 83).
2. Perform one of the following tasks:
   - To add a join, select the table in the minischema to which you want to add a join, right-click, and select Add Joins.
     The table you selected is displayed in the left Table drop-down list box of the Edit Properties of Minischema Join dialog box.
     
     Note:
     Because you can only create self joins in minischemas created from text file data sources, the selected table displays in both the left and right Table text box, and neither is editable.
     
     Alternatively, you can right-click in the background of the minischema and select Add Joins, but in this case, no table names are displayed in either of the Table drop-down list box.
     
     Note:
     For self joins in minischemas created from text file data sources, only the left Table drop-down list box is active. The right Table text box displays the name of the table you select in the left Table drop-down list box.
   - To edit a join, double-click the appropriate join line in the minischema to display the Edit Properties of Minischema Join dialog box.
     
     Note:
     For self joins, the left and right Table text box is not selectable.
3. In the right Table drop-down list box, select the table to which you want to establish a join.
   
   Note:
   For minischemas built from text file data sources, you can only perform self-joins; therefore, the Table drop-down list box is not selectable.
4. If you are adding a join, click the + button.
Note:

If you are editing a join, or adding the first join, skip this step and proceed to step 5.a.

Alternatively, you can delete joins from the list by selecting the join pair from the Column grid and clicking the red “X” button.

5 To work with joins from a relational data source:
   a. In the Column grid, point the mouse into the left cell of the first empty row in the grid and click to display a list of the column names associated with that table.
      Alternatively, if you are editing an existing join, click in the left cell of the join pair that you want to edit to display the list of column names.
   b. Select a column to begin the join.
   c. In the Column grid, point the mouse into the right cell of the same row of the grid and click to display a list of the column names associated with that table.
      Alternatively, if you are editing an existing join, click in the right cell of the join pair that you want to edit to display the list of column names.
   d. Select a column to complete the join.
   e. Optional—if this is an outer join, select the Outer check box and then choose one of the following options:
      ● Left
      ● Full
      ● Right
   f. Optional—Repeat step 2 through step 5.e for each new join combination you want to create.
   g. Optional—Repeat step 4 through step 5.e for each join you want to add for the currently selected pair of tables.

6 To work with self joins from a text file data source:
   a. In the left Column drop-down list, select the column name to begin the join.
   b. In the right Column grid, select the column name to complete the join.
   c. Optional—if this is an outer join, select the Outer check box and then choose one of the following options:
      ● Left
      ● Full
      ● Right
   d. Optional—Repeat step 2 through step 6.c for each new self join pair you want to create.
   e. Optional—Repeat step 5.a through step 6.c for each self join you want to add for the currently selected table.
To save your join additions and edits, right-click in the minischema work area and select Save Minischema.

**Note:**
You can also add joins by using the mouse to draw the joins between the appropriate columns of two tables in the minischema. Then, double-click the join line to open the Edit Properties of Minischema Join dialog box where you can edit the join properties.

### Adding Joins By Inspection

You can have Essbase Studio perform an inspection of your database and return a list of suggestions of possible join pairs.

To add joins by inspection:

1. If not already opened, open the minischema (see “Opening a Minischema” on page 83).
2. Right-click in the background of the minischema and select Add Joins By Inspection.
   
   The Create Joins by Inspection dialog box is displayed listing the possible join pairs that were detected by Essbase Studio.
3. **Optional**—To filter the results by table or by column, perform either of these tasks:
   - Type the name of a table to filter on in **Table name filter**.
     
     Table names are case-sensitive.
     
     If the table name you entered is contained in either the Join Source or Join Target side of the join pair, those pairs are displayed.
   - Type the name of a column to filter on in **Column name filter**.
     
     Column names are case-sensitive.
     
     If the column name you entered is contained in either the “Join Source” or “Join Target” side of the join pair, those pairs are displayed.
4. Perform one of the following tasks to choose the join pairs you want to add to the minischema:
   - To choose all the join pair results, select the **Select all items** check box.
   - To choose selected join pair results, select the check box next to the join pair in the results grid.
5. Click **OK** to accept your join pair selections.
6. Verify that the selected join pairs were added in your minischema diagram.
7. To save your changes, right-click in the minischema work area and select Save Minischema.

### Adding Tables in a Minischema

You can modify a minischema by adding tables to it from a physical source.
You can follow the dialog box-based procedure described in “Adding or Removing Tables in a Minischema” on page 81 or use the drag-and-drop process described below.

➤ To add tables to a minischema:

1 If not already opened, open the minischema (see “Opening a Minischema” on page 83).

2 In the Source Navigator, select the Data Sources tab, and then expand the data source connection name from which you want to add a table.

   A list is displayed of all the physical table objects that you added to the data source during the data source creation process.

3 Select the table you want to add and drag it into the minischema work area.

   A join will be made automatically if it is already present in the physical data source.

   Note:

   You cannot add a physical table to the minischema more than once.

4 Repeat step 3 for all physical tables you want to add to the minischema.

5 To save your changes, right-click in the minischema work area and select Save Minischema.

Viewing Sample Data

You can view the available sample data of a table in a minischema or from one or more individual columns.

➤ To view sample data in a minischema:

1 If not already opened, open the minischema (see “Opening a Minischema” on page 83).

2 Perform either or both of the following tasks:

   ● To view data for a table in the minischema, right-click the table and select View Sample Data.

   You can select the table from either the minischema work area or from the Minischema tab in the Source Navigator.

   ● To view data for a single column from a table in a minischema, right-click the appropriate column and select View Sample Data.

   Alternatively, to view data for one or more columns within a table, press and hold the Ctrl key, click on the columns whose data you want to view, then right-click and select “View Sample Data.”

   You can select columns from either the minischema work area or from the Minischema tab in the Source Navigator.

   The sample data you requested is displayed in a new tab in the work area of the console.

3 To close the sample data window, click the X in the tab for this window.
Applying Color to Minischema Tables

You can use color to denote the various tables in a minischema. For example, you may have a very large minischema with multiple fact tables. You can set the fact tables to appear as red to make them easy to locate in the minischema work area.

You can also change previously-set colors in a minischema.

When you apply color to a minischema table, the color is applied to the top tab portion of the table element.

➤ To apply color to minischema table:

1 If not already opened, open the minischema (see “Opening a Minischema” on page 83).
2 Select a table element to which to add color.
   Alternatively, press and hold the Ctrl key, then click on all the tables to which you want to color.

   Note:
   You can choose tables that have no color added to them (these appear as light gray) or tables that were previously colored.

3 Right-click and select Color to view a list of available colors from which you can choose.
4 Select a color from the list.
5 View the effect on the minischema in the minischema work area.
6 To save your color changes, right-click in the minischema view and select Save Minischema.

Creating Metadata Elements from Minischema Objects

By performing a simple drag-and-drop operation, you can create metadata elements from objects in a minischema. This can be a quick and easy way to create alternate views of a data source or of certain elements from a source.

➤ To create metadata elements from minischema objects:

1 If not already opened, open the minischema (see “Opening a Minischema” on page 83).
2 Select one or more objects in the minischema from which you want to create a metadata element. For example:
   ● To create a metadata element from one table in the minischema, select the table, right-click, and then select Add to Metadata Navigator.
   ● To create a metadata elements from multiple tables in the minischema, hold the Shift key down, then click on the appropriate tables to select them, right-click, and then select Add to Metadata Navigator.
   ● To create a metadata element from one column in the minischema, select the column, right-click, and then select Add to Metadata Navigator.
To create a metadata elements from multiple column in a minischema table, hold the Ctrl key down, then click on the appropriate columns in the minischema table to select them, right-click, and then select Add to Metadata Navigator.

After performing one of the actions listed above, the “Choose Folder” dialog box is displayed.

3 In Choose Folder, navigate to the folder in which you want to store the metadata element, and then click OK.

Note:

At this time, you can create a new folder in which to store the metadata element. Click the “Create Folder” button in the “Choose Folder” dialog box and specify a “New Folder Name” in the popup dialog box. After you click OK, navigate to the new folder in the “Choose Folder” dialog box, and then click OK.

4 Verify that the minischema element you specified has been added to the Metadata Navigator.

5 Repeat step 2 and step 3 for all metadata elements you want to create.

**Viewing Minischemas**

Because minischema diagrams can be very large at times, Essbase Studio Console provides tools and commands to help you to navigate them.

- Use the thumbnail viewer to navigate in a thumbnail size view of your minischema to the point at which you want to focus.

  See “Using the Minischema Work Area” on page 89.

- Maximize the minischema work area to gain the most screen real estate. Used in conjunction with the thumbnail viewer, you can quickly pinpoint areas of the minischema work area on which you want to focus.

  See “Maximizing and Minimizing the Minischema Work Area” on page 92.

- Use the zoom in and zoom out commands to enlarge or decrease the size of the minischema elements in your work area.

  See “Zooming In and Zooming Out on the Minischema Work Area” on page 93.

**Using the Minischema Work Area**

Use the thumbnail viewer to pinpoint an area in the minischema work area on which you want to focus.

Note:

The thumbnail viewer is available in both the minimized and maximized views of the minischema in the work area (see “Maximizing and Minimizing the Minischema Work Area” on page 92).
To use the thumbnail viewer:

1. If not already opened, open the minischema (see “Opening a Minischema” on page 83).

2. To launch the thumbnail viewer, click the viewer icon, [], in the bottom right of the minischema work area.

   The thumbnail viewer pops up in the lower right corner of the minischema work area. The thumbnail viewer contains the same portion of the minischema as displayed in the main minischema work area, but in a miniature format. A smaller, transparent blue pointer covers a portion of the minischema.

   Figure 2  Thumbnail Viewer

3. Drag the pointer in any direction to the location in the minischema work area that you want to view.
Click the X in the top right corner of the thumbnail viewer to close it.

The focus of the minischema work area is now on the section you selected in the thumbnail viewer.
Maximizing and Minimizing the Minischema Work Area

In the default layout of the Essbase Studio Console, the minischema work area is displayed in the minimized format, between the Metadata Navigator and Source Navigator. You can expand the minischema work area to help you to navigate your way through large minischema diagrams.

Note:

Both the minimized and maximized minischema work areas support the thumbnail viewer (see “Using the Minischema Work Area” on page 89).
To maximize the minischema work area, click the Maximize button,  
, in the top right corner of the minischema work area.

To minimize the minischema work area, click the Minimize button, , in the top right corner of the minischema work area.

**Zooming In and Zooming Out on the Minischema Work Area**

You can change the view size of the minischema objects by using the zoom in and zoom out commands. Zoom in to enlarge the size of the objects. Zoom out to reduce the size of the objects.

To enlarge the size of the objects in a minischema, right-click in the minischema work area and select **Zoom In**.

To reduce the size of the objects in a minischema, right-click in the minischema work area and select **Zoom Out**.

The Zoom In and Zoom Out commands may be repeated as many times as necessary to obtain the required view size.

**Arranging the Tables in a Minischema**

You can manually place the tables in a minischema wherever you want in the minischema work area. Alternatively, Essbase Studio can arrange the tables for you.

To automatically arrange, or lay out, the tables in a minischema, with the minischema displayed in the minischema work area, click the **Lay out schema** button,  .

**Note:**

You can also automatically arrange the tables by right-clicking in the minischema work area.

**Editing Properties of a Minischema**

When you edit the properties of a minischema, you are modifying the properties that you specified when you completed the Add/Remove programs dialog box in the Minischema Wizard.

The editing tasks you can perform are:

- Change the data source
- Add or remove tables from the minischema
- Create a filter
- Add related tables
To edit minischema properties:

1. **Optional:** If not already opened, open the minischema (see “Opening a Minischema” on page 83).
2. From the Source Navigator, select the Minischema tab.
3. Under Minischemas in the physical tree, right-click the minischema to edit and select Minischema Properties.
4. Follow the instructions for editing minischema properties in “Adding or Removing Tables in a Minischema” on page 81.

**Note:**
The instructions for editing minischema properties are the same procedures documented in “Adding or Removing Tables in a Minischema” on page 81.

### Refreshing the Minischemas List

To refresh the list of minischemas in the Minischema tab of the Source Navigator, right-click the top item in the tree, “Minischema,” and select Refresh.

### Deleting Minischemas

To delete an existing minischema, in the Minischema tab of the Source Navigator, right-click the name of the minischema you want to delete and select Delete.
Introspection Overview

Introspection is a method of inspecting a physical data source for metadata elements. When you perform introspection, structural information that already exists in the data source is inspected to detect fact tables, dimension tables, hierarchies, aliases, and attributes. The metadata elements derived from introspection are then used to create cube schemas and, optionally, Essbase models.

You can perform introspection during the data source creation process. Or you can perform introspection on an existing data source to “scrape” the data source for the most up-to-date information.

Note:

Introspection is not supported for Oracle Business Intelligence or text file data sources.

When you choose to perform introspection, the process analyzes the data source and detects and presents candidates for the following areas:

- Fact table—as described in “Selecting a Fact Table in the Introspection Wizard” on page 96
- Dimension tables—as described in “Selecting Dimension Tables in the Introspection Wizard” on page 97
- Hierarchies—as described in “Selecting Hierarchies in the Introspection Wizard” on page 97

You also have the option of creating a minischema with the information collected.

The Introspection Wizard guides you through the process. The wizard allows you view the candidate selections and choose which items to keep or reject. The process begins with selecting a minischema option.
Selecting a Minischema Option in the Introspection Wizard

Begin the introspection process by selecting a data source to examine then choosing a minischema option.

➤ To begin the introspection process and choose a minischema option:

1 In the Source Navigator, right-click the data source on which you want to perform introspection and select Introspect.

2 Choose an option:
   ● Create a new schema diagram—if you choose this option, enter a name for the new minischema
   ● Reuse an existing minischema diagram—if you choose this option, select the existing minischema to reuse. Also, if you want to clear the content of the existing minischema before you begin, select the Clear existing schema check box.
   ● Skip creation of a minischema diagram—if you choose this option, you are still guided through the introspection process, but you will not create a minischema diagram.

   Note:
   If you are accessing a text file data source, you may create minischema, but it will not contain joins.

3 Ensure that Use Introspection to Detect Hierarchies is selected.

4 Click Next; or, if you chose not create a minischema, click Finish.

Clicking Next takes you to the Select Fact Tables page of the wizard, as described in “Populating a Minischema for Relational Sources” on page 57.

Selecting a Fact Table in the Introspection Wizard

During the introspection process, you are presented with fact table candidates, from which you make a selection.

➤ To select a fact table:

1 In Select Fact Table(s), select the objects you want to use as fact tables, and click Next.

The Fact Table(s) list should already contain tables. These are tables which, after examining the data source, Essbase Studio proposes as possible fact tables.

You can accept one or more of these selections or, using the arrow keys, remove them and choose a table from Available Tables to use as the fact table.

   Note:
   The Available Tables list contains all the remaining tables, views, alias tables, and synonyms that were selected when the data source was created.
You can choose multiple tables to use as fact tables.

2 Optional—Enter a Filter to limit the tables displayed in Available tables, and click Apply.

For example, if database tables are prefixed for a certain business area, such as “perf” for Performance, enter a filter to return only those tables in that area, such as:

perf*

Filters apply to all tables, views, aliases, or synonyms you have chosen to display

3 Click Next to view the Select Dimension Table(s) page of the Introspection Wizard.

**Selecting Dimension Tables in the Introspection Wizard**

During the introspection process, you are presented with dimension table candidates, from which you make selections.

➤ To select dimension tables:

1 In Select Dimension Table(s), select the objects you want to use as dimension tables,

   The Selected Dimension Table(s) list may already contain objects. These are objects which, after examining the data source, Essbase Studio proposes as possible dimension tables. You can accept these choices or, using the arrow keys, remove them and choose one or more objects from Available Objects to use as dimension tables.

2 Optional—Enter a Filter to limit the tables displayed in Available tables, and click Apply.

   For example, if database tables are prefixed for a certain business area, such as “perf” for Performance, enter a filter to return only those tables in that area, such as:

   perf*

   Filters apply to all tables, views, aliases, or synonyms you have chosen to display

3 Click Next to view the Select Hierarchies page of the wizard.

**Selecting Hierarchies in the Introspection Wizard**

➤ To select hierarchies to add as metadata elements during introspection:

1 In Select Hierarchies, select the hierarchies you want to use from the Available Hierarchies list.

   Available Hierarchies presents the hierarchies discovered by Essbase Studio after examining the data source. All hierarchies are selected by default. You can do any of the following:

   ● Accept all the hierarchies discovered.

   ● Select the hierarchies that you do not want to add as metadata elements and click the Delete selected item button.

   ● Select a hierarchy within a multi-chain hierarchy and click the Delete selected item button.
● Select individual members in a hierarchy and click the Delete selected item button.

● Build a hierarchy by selecting the Create new hierarchy button. Then, follow these steps:
  a. In the Edit Hierarchy dialog box, enter a Name for the hierarchy and an optional Description.
  b. Optional—If this is a measures hierarchy, select the Create as measures hierarchy check box.
  c. Click OK to return to the Select hierarchies page of the Introspection Wizard.
  d. Select the new hierarchy in the Hierarchies list, then navigate in the Available Tables and Columns list to the first column to add to the hierarchy, select it, and click the Add column as child button.

  Note:
  The first column you add to the hierarchy must be added as a child.
  e. Add other columns to the hierarchy, using either the Add column as child or Add column as sibling button.

● Build a hierarchy by selecting a table in the Available Tables and Columns and clicking the Add hierarchy for table button. Then, follow these steps:
  a. Select the new hierarchy in the Hierarchies list.
     The format for the hierarchy name is: tablenameHierarchy.
  b. Navigate in the Available Tables and Columns list to the first column to add to the hierarchy, select it, and click the Add column as child button.
  c. Add other columns to the hierarchy, using either the Add column as child or Add column as sibling button.

2 Click Finish.

If you chose to create a new minischema, you can review it in the Minischema work area.

Also, review the metadata elements created as a result of the introspection process. They are listed in the Metadata Navigator under the appropriate folder.
Metadata Elements Overview

Metadata elements are the logical objects derived from the physical objects in a data source when you create a data source connection. Metadata elements can also be created from other metadata that is already stored and cataloged for Essbase Studio use. There are several types of metadata elements for specific purposes. These metadata elements are:

- Dimension elements—a logical representation of a physical source column.
  - Dimension elements can be created during the data source connection process or at any time after.
  - Dimension elements can be text, numeric, or date based.
  - Dimension elements can be edited to specify bindings other than the physical column on which they are based.
  - Dimension elements can be edited to add a filter.
  - A sort order can be added to a dimension element.
- Derived text measures—a text measure whose values are governed by a predefined rule expressed as a range.
- Date elements—Granular date-type metadata elements that Essbase Studio derives from an existing date-type metadata element.

The metadata elements you create can be used to build other metadata elements, such as hierarchies, measure hierarchies, and calendar hierarchies. Hierarchies are described in Chapter 9, “Hierarchies”.

A folder is another type of metadata element. Folders aid in the organization of items in the Metadata Navigator. See “Creating or Editing Metadata Folders” on page 109.
Creating or Editing Metadata Elements

See the following sections for instructions on creating metadata elements:

- “Creating or Editing Dimension Elements and Derived Text Measures” on page 100
- “Creating or Editing Derived Text Measures” on page 107
- “Creating Date Elements” on page 109
- “Creating or Editing Metadata Folders” on page 109

Creating or Editing Dimension Elements and Derived Text Measures

The workflow for creating or editing dimension elements or derived text measures is:

- For dimension elements:
  - Create the dimension element.
    See “Creating Dimension Elements” on page 100.
  - Define or edit the expression, filter, binding, and sort order properties in the Main tab of the Edit Properties dialog box.
    See “Defining or Editing General Properties for Dimension Elements” on page 102.

- For derived text measures, create or access the derived text measure, then define the expression and range values in the Main tab of the Edit Properties dialog box.
  See “Creating or Editing Derived Text Measures” on page 107.

- For dimension elements and derived text measures, edit any applicable alias sets bindings.
  See “Creating or Editing Alias Set Bindings for a Given Alias Set” on page 108.

Creating Dimension Elements

There are several methods for creating dimension elements as described in the following procedures:

- Creating a Dimension Element from the Metadata Navigator
- Creating a Dimension Element from the Data Sources Tab or Minischemas Tab
- Creating a Dimension Element from a Minischema

Creating a Dimension Element from the Metadata Navigator

➤ To create a dimension element from the Metadata Navigator:

1 In the Metadata Navigator, right-click on the folder where you want to store the dimension element, and select New > Dimension Element.
   This opens the Edit Properties dialog box for the dimension element.
2 Enter a Name and optional Description.

3 Proceed to “Defining or Editing General Properties for Dimension Elements” on page 102 to define the binding expression and, optionally, filters and sort order.

Creating a Dimension Element from the Data Sources Tab or Minischemas Tab

➤ To create a dimension element from the Data Sources tab:

1 In the Data Sources tab, navigate to the physical element upon which you want to base the dimension element.

2 Select the element and drag it from the Data Sources tab and drop in on the folder where you want to store it in the Metadata Navigator.

Note:
You can select table or column elements from the Data Sources tab.

3 Proceed to “Defining or Editing General Properties for Dimension Elements” on page 102 to define the binding expression and, optionally, filters and sort order.

➤ To create a dimension element from the Minischemas tab:

1 Navigate in the Minischemas tab to the physical element upon which you want to base the dimension element.

2 Select the element and drag it from the Minischemas tab and drop in on the folder where you want to store it in the Metadata Navigator.

Note:
You can select table or column elements from the Minischema tab.

3 Proceed to “Defining or Editing General Properties for Dimension Elements” on page 102 to define the binding expression and, optionally, filters and sort order.

Creating a Dimension Element from a Minischema

➤ To create a dimension element from the minischema:

1 In the Minischemas tab, navigate to the minischema which you want to use to base dimension elements.

2 Right-click the minischema and select Open Minischema.

You can also double-click the minischema to open it.

3 In the minischema work area, drag an element from the diagram and drop in on the folder where you want to store it in the Metadata Navigator.

Note:
You can also create a dimension element from the minischema by selecting table or column elements from the minischema diagram and selecting Add to Metadata Navigator.
Defining or Editing General Properties for Dimension Elements

Before you define or edit the general properties for dimension elements, create a dimension element using any of the methods described in “Creating or Editing Metadata Elements” on page 100.

Derived text measures must be created and edited using the method described in “Creating or Editing Derived Text Measures” on page 107.

Note:
Caption bindings, key bindings, filters, and sort elements must all come from the same data source.

To define the general properties for dimension elements:

1. Perform one of the following actions:
   - To create a dimension element, complete one of procedures described in “Creating Dimension Elements” on page 100.
   - To edit a dimension element, locate it in the Metadata Navigator, right-click, and select Edit.

   Note:
   You can also open the dimension element by double-clicking it.

2. In the Edit Properties dialog box, create an expression on which to base the dimension.
   For example, the following expression bases the dimension element on the SKU column from the Product table:
   
   `connection : \'tbc_source\'::\'tbc.product\'\.'SKU\'

   You may also enter (hand type) an expression or statement of your own. For example, you may want to enter the string “Seasons” to use as the member name that displays in the Essbase outline or spreadsheet for this element.

   Note:
   The expression on which the dimension element is based is written in CPL (Common Platform Language). The expression is a sequence of operands and operators following the language-defined syntax. Each expression returns a value, the type of which defines the type of the expression. See Appendix D, “CPL Reference”.

Proceed to “Defining or Editing General Properties for Dimension Elements” on page 102 to define the binding expression and, optionally, filters and sort order.
Note:
If you enter (hand type) an expression or statement of your own to define or edit a dimension element, you must follow the guidelines for delineating strings and names. See “Entering User-Defined Expressions” on page 106.

Note:
For caption binding and key binding expressions, text file data sources only support the substring (substr) function and the concatenation operator ( || ).

Note:
Essbase Studio does not verify cycle dependencies between metadata elements. Cycle dependency is a relationship between metadata elements that cycles back upon itself. See “Cycle Dependency Guidelines” on page 265 for more information.

Note:
For limitations on creating expressions, see “Independent Dimension Bindings Limitations” on page 267

Follow these steps to create the expression:

a. Select the Source tab in the lower-left of the dialog box.
b. Expand the dimensions to display the members.
c. Select a member.
d. Use the right-direction arrow to move the connection string for the member to the Caption Binding text box.
e. Select the Functions tab.
f. Expand the SQL level to display the function types.
g. Expand the function types to display the functions.
h. Select a function.
i. Use the right-direction arrow to move the function string to the Caption Binding text box.
j. Select the Operators tab.
k. Expand the operator types to display the operators.
l. Select an operator.
m. Use the right-direction arrow to move the operator to the Caption text box.

3 Choose a Key Binding option for this dimension element depending on the type of outline in which this dimension element is going to be used:
● Same as caption—This is the default. Essbase Studio uses the expression you input in step 2 to indicate how to load member names for this dimension element into the Essbase outline or in a spreadsheet.

**Tip:**
Use the “Same as caption” binding option when the instance of this dimension element has a unique name and when you know that this particular dimension element will participate in an outline that contains unique member names.

When you choose “Same as caption,” the expression in the Caption Binding text box is copied to the Key Binding text box.

● Advanced—Select this option and then enter an expression in the **Key Binding** text box that tells Essbase how to build this member.

Use the “Advanced” binding option when you know that the Caption Binding contains duplicate member names and you know how to build the unique member key from the data source.

**Tip:**
Use the “Advanced” binding option when this particular dimension element will participate in an outline that contains duplicate member names.

**Note:**
For dimension elements with a data type other than text (string), the Advanced option is set automatically.

In the Key Binding text box, you provide the expression that generates the unique identifier for each value passed to Essbase for this metadata element. The expression you provide should be associated with one or more member key columns from the data source. Essbase Studio makes use of the assigned keys to uniquely identify the members that will be associated with this dimension element.

For example, suppose you know that in the “Market” table in your data source, there is a column called “cityNames,” which contains duplicate member names. Another column, “cityID,” is the member key column for the Market table and each city has a unique city ID in this column. Because of this, cityID would be a good candidate to use for the key binding expression. Or, using the concatenation operator, you may prefix the cityName column with the region name from another column in the table, “Region,” to create the key binding expression.

Drag the appropriate elements from the Source, Functions, and Operators tabs in the Formula box and drop them in the Key Binding text box.
**Note:**

For caption binding and key binding expressions, text file data sources only support the substring (substr) function and the concatenation operator (||).

When you choose “Advanced,” the expression in the Key Binding text box will probably differ from the expression in the Caption Binding text box.

- **Delayed**—Select this option to allow Essbase Studio to automatically generate a key binding, or to provide the key binding expression yourself at a later time.

**Tip:**

Use the “Delayed” option when you are not sure how to generate the key binding expression. Essbase Studio can do this for you.

You can let Essbase Studio generate the binding expression for you or provide the expression during hierarchy creation or editing.

You may also return to edit a dimension element already used in a hierarchy and change the key binding option to “Delayed.” You may then choose to let Essbase Studio generate the default keys automatically in all the hierarchies in which the dimension element participates. Or, you may specify them yourself during the hierarchy editing process.

In the hierarchy, Essbase Studio generates the full path for the key binding. For example, in the following hierarchy, the key binding for each dimension element is the default caption binding:

- FAMILY - connection : \\tbcSource::tbc.family::FAMILY
- SKU - connection : \\tbcSource::tbc.product::SKU

If the key binding is changed to Delayed for the SKU dimension element, the key binding in the hierarchy, generated by Essbase Studio, is the full path of the dimension element in relation to its place in the hierarchy:

- class : \\tbcSource\'\productdim\'\FAMILY\'\caption || "_" || class : \\tbcSource\'\product\'\SKU\'\caption

The Delayed option can be used for Essbase outlines that support either unique or duplicate member names,

When you choose the Delayed option, the Key Binding text box is left blank.

4 **Optional**—To create a filter for this dimension element, drag the appropriate elements from the Source, Functions and Operators tabs in the Formula box and drop them in the Filter box, and then type the filter criteria as needed.

**Note:**

The steps for creating a filter are the same as those steps used to create an expression on which to base the dimension. See step 2 on page 102
For example, if you know that all diet products have SKU in the format, XXX-20, then a filter can be set using the `substr()` function as follows:

```
'substr'( connection : 'tbc_source'::'tbc.product'.'SKU', 5, 2 ) == "20"
```

**Note:**

For filter expressions, text file data sources only support the substring (substr) function and these operators: >, >=, <, <=, |=, ==.

5 Optional—To define the sort order of columns in the dimension, drag the appropriate column from the Source tab in the Formula list and drop it under Sort Column; then, click the down arrow in the Sort Order column to choose the Ascending or Descending option.

**Note:**

If you do not see a down arrow in the Sort Order column, click in the row for the appropriate column under Sort Order to activate the Ascending/Descending options.

6 Optional—Associate any applicable alias sets with the dimension element, as described in “Creating or Editing Alias Set Bindings for a Given Alias Set” on page 108.

**Entering User-Defined Expressions**

If you enter (hand type) an expression or statement of your own to define or edit a dimension element, you must follow the guidelines for delineating names and strings. This topic discusses:

- “Entering Names in User-Defined Expressions” on page 106.
- “Entering Strings in User-Defined Expressions” on page 106

**Entering Names in User-Defined Expressions**

If you enter (hand type) an expression or statement of your own to define or edit a dimension element, you must follow these guidelines for delineating names:

- A name must be preceded and followed by single quotes. Here are two examples:
  - The name `ab` must be entered as `'ab'`
  - The name `a"b` must be entered as `'a"b'`
- Single quotes within a name must be repeated; for example:
  - The name `a 'b` must be entered as `'a 'b'`
- Single quotes preceding or following a name must be preceded and followed by two single quotes; for example:
  - The name `a 'b` must be entered as `''a ''b''`

**Entering Strings in User-Defined Expressions**

If you enter (hand type) an expression or statement of your own to define or edit a dimension element, you must follow these guidelines for delineating strings:

- A string must be preceded and followed by double quotes. Here are two examples:
The string \texttt{ab} must be entered as "\texttt{ab}"

The string \texttt{a\'b} must be entered as "\texttt{a\'b}"

- Double quotes within a string must be repeated; for example:
  
The name \texttt{a"b} must be entered as "\texttt{a"b}"

- Double quotes preceding or following a string must be preceded and followed by two double quotes; for example:
  
The name "\texttt{a"b}" must be entered as ""\texttt{a"b}""

### Creating or Editing Derived Text Measures

Derived text measures are text measures whose values are governed by a predefined rule expressed as a range. For example, a derived text measure, called "Sales Performance Index," based on a measure Sales, could consist of the values "High," "Medium," and "Low." This derived text measure is defined to display "High," "Medium," and "Low" depending on the range in which the corresponding sales values fall.

To create or edit a derived text measure:

- Define the formula and range in the Main tab of the Edit Properties of Dimension Elements dialog box, as described in the procedure in this section.

- Associate any applicable alias sets with the dimension element.
  
  See “Creating or Editing Alias Set Bindings for a Given Alias Set” on page 108.

**Note:**

When derived text measures are used in cube schemas to build Essbase models, the “XOLAP Model” option will not be available for the model.

➤ **To create a derived text measure:**

1. In the Metadata Navigator, right-click on the folder where you want to store the derived text measure and select **New > Derived Text Measure**.

2. In the **Properties** dialog box, enter a **Name** and optional **Description** for this derived text measure.

3. To create the expression on which to base the derived text measure, drag the appropriate numeric source column from the list of tables and columns in the **Formula** list and drop it in the **Expression** box.

   The Functions and Operators tabs and their corresponding elements are not available for derived text measures.

**Note:**

You cannot create expressions for derived text measures from elements in text file data source connections.
Creating or Editing Alias Set Bindings for a Given Alias Set

You can create alias set bindings in the Alias Set Manager. You can also create or edit alias set bindings when creating or editing dimension elements or derived text measures.
Note:

Use care when adding or deleting alias set bindings. Any bindings you add or delete in this dialog box are directly reflected in the bindings for the selected alias set in the Alias Set Manager.

To create or edit the alias set bindings for a given alias set:

1 If the element with which you want to work is not already open, perform one of the following actions:
   - To create a new dimension element in the Metadata Navigator, right-click on the folder where you want to store the dimension element, and select New > Dimension Element.
   - To edit a dimension element, locate it in the Metadata Navigator, and select Edit.

This opens the Edit Properties dialog box for the dimension element.

2 Select the Alias tab.

3 To create or edit an alias binding expression, click the Edit Alias Set button.

4 In the Alias Bindings dialog box, select the column that contains the alias information which you want to bind to this element, and click Apply.

5 Optional—To delete an alias binding, select the appropriate row in the Alias Sets grid and click the Delete button.

6 Repeat steps step 3 and step 4 for each alias binding you want to create or edit.

7 When finished, click OK to close the dialog box.

Creating Date Elements

To create date elements:

1 In the Metadata Navigator, locate the logical date column on which you want to base new date elements.

   The logical date column you select must represent a date column in your physical data source. Otherwise, the Create Date Elements function is not available.

2 Right-click the logical date column and select Create Date Elements.

3 In the Create date elements dialog box, select the date elements you want to create from the list of available date elements.

4 Click OK.

   In the Metadata Navigator, the new date elements appear as text columns belonging to the same table as the original date element.

Creating or Editing Metadata Folders

At data source creation time, if you choose to create metadata elements, Essbase Studio provides a folder to store the elements in the Metadata Navigator, and gives a default name to the folder.
However, you can choose to store elements in folders you designate. Further, you can change the names and descriptions of these folders at any time.

➤ To create a new metadata folder in the Metadata Navigator:
1. Navigate to the location where you want to create the folder, and select File > New > Folder.
2. Enter a folder Name.
3. Enter an optional Description.
4. Click Apply, then click OK.

➤ To edit a folder name or description:
1. Right-click the appropriate folder name and select Edit.
2. Update the Name.
3. Optional—Update the existing Description.
4. Click Apply, then click OK.

Working with Metadata Elements

You can perform these tasks with metadata elements:

- “Copying Metadata Elements” on page 110
- “Renaming Metadata Elements” on page 111
- “Deleting Metadata Elements” on page 111
- “Viewing Sample Data for Metadata Elements” on page 111
- “Showing Lineage” on page 111

Copying Metadata Elements

Under construction.

➤ To copy a metadata element to another location:
1. In the Metadata Navigator, locate the metadata element to copy.
2. Right-click the element and select Copy.
3. Navigate to location to which you want to copy the element, either the root or a folder, right-click, and select Paste.
Renaming Metadata Elements

➤ To rename a metadata element, perform one of the following actions in the Metadata Navigator:
  ● Double-click the element to launch the element's properties dialog box.
  ● Right-click the element and select Rename, then type the new name.

Note:
When renaming metadata elements, you must follow the guidelines in Appendix C, “Naming Restrictions for Essbase Studio”.

Deleting Metadata Elements

➤ To delete a metadata element, right-click the element name in the Metadata Navigator and select Delete.

Viewing Sample Data for Metadata Elements

➤ To view the sample data for a given a metadata element, right-click the element name in the Metadata Navigator and select View Sample Data.

Showing Lineage

➤ To view the lineage for a given a metadata element, right-click the element name in the Metadata Navigator and select Show Lineage.

For information on lineage, see Chapter 14, “Lineage”
Alias Sets

In This Chapter

Alias Set Overview ............................................................................................................. 113
Working with Alias Sets..................................................................................................... 113

Alias Set Overview

The use of alternate names, or aliases, to identify objects is a common business practice. In Essbase, an alias is an alternate name for a member or shared member. For example, members in the Product dimension in the Sample Basic database are identified both by product codes, such as 100, and by more descriptive aliases, such as Cola. Aliases are stored in alias tables. Aliases can improve the readability of an outline or a report. Aliases can be grouped by languages, regions, or descriptive names.

Using alias sets, you can specify more than one alias for a member. For example, you could use different aliases for different kinds of reports—users may be familiar with 100-10 as Cola, but advertisers and executives may be familiar with it as The Best Cola. This list shows some products in the Sample Basic database that have two descriptive alias names:

<table>
<thead>
<tr>
<th>Product</th>
<th>Default</th>
<th>Long Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>100-10</td>
<td>Cola</td>
<td>The Best Cola</td>
</tr>
<tr>
<td>100-20</td>
<td>Diet Cola</td>
<td>Diet Cola with Honey</td>
</tr>
<tr>
<td>100-30</td>
<td>Caffeine Free</td>
<td>Cola All the Cola, none of the Caffeine</td>
</tr>
</tbody>
</table>

You create alias sets in the Alias Set Manager. An alias set maps a specific, named set of alias names to member names. For each element in the logical tree, you assign one alias for each alias set defined in the catalog. Unlike Essbase, there is no default alias table. You specify all alias tables, including a Default alias table if you choose, in the Alias Set Manager. You then associate alias sets to members in an Essbase model before the cube deployment process in these ways:

- By applying one or more alias sets to all members at the Essbase model level
- By applying one or more alias sets at the dimension and member level

For more information, see “Working with Alias Sets” on page 113.

Working with Alias Sets

Use the Alias Set Manager to:

- Create alias sets—see “Creating Alias Sets” on page 114
Creating Alias Sets

Alias sets are the container that holds the alias bindings that you create.

To create an alias set:

1. In the Essbase Studio Console, select Tools > Alias Set Manager.
2. To the right of the Alias sets list box, click the Create an alias set button.
3. At the prompt, enter a name for the alias set, and click OK.
   The name you entered now appears in the “Alias sets” list box in the Alias Set Manager.
4. Optional—Enter a Description for the alias set.
   You can now create one or more bindings for this alias set, as described in “Creating Bindings for an Alias Set” on page 114.

Creating Bindings for an Alias Set

Once you have created an alias set (as described in “Creating Alias Sets” on page 114), you can add bindings to it. Bindings specify the column or columns in the data source which hold alias information for a particular alias set.

To create bindings for an alias set:

1. If you have not already done so, open the Alias Set Manager (Tools > Alias Set Manager).
   Before you can add bindings to an alias, you must have already created an alias set, as described in “Creating Alias Sets” on page 114. If you have already created an alias set, proceed to step 2.
2. In the Alias sets list box, select the alias set for which you want to add bindings.
3. To the right of the Bindings list box, click the Create a binding button to display the Create Binding dialog box.
   The Create Binding dialog box consists of the following sections:
   - The Binding source tree, which displays all items in the metadata tree that can have an alias binding associated with them.
   - The Binding expression text box, where the binding expression is entered.
     Use the source tree below the Binding expression text box to generate a binding expression for the metadata object you selected in the “Binding source” tree. Alternatively, you can type an expression direction into this text box.
4. In the Binding source tree, select a metadata element for which you want to associate an alias binding.
In the source tree under the Binding expression text box, navigate to the column which you want to use to generate a binding expression.

Double-click the column name to add the binding expression to the text column.

Click OK.

The expression you created is displayed in the Bindings section of the Alias Set Manager in the following format:

- Under Source element, the metadata element which you are associating with an alias is displayed.
- Under Binding, the source element on which you are basing the alias is displayed.

Repeat step 3 through step 7 for all bindings you want to create for the selected alias set.

When you have created all bindings for an alias set, click OK to exit the Alias Set Manager.

Editing Alias Sets

You can perform these editing functions on the bindings in alias sets:

- Modify an existing binding in the set—see “Modifying Bindings in an Alias Set” on page 115
- Add bindings to the set—see “Adding Bindings to an Alias Set” on page 116
- Delete bindings from the set—see “Deleting Bindings from an Alias Set” on page 116

Modifying Bindings in an Alias Set

You can modify an existing binding in an alias set.

To modify a binding:

1. In the Essbase Studio Console, select Tools > Alias Set Manager.
2. Under Alias Sets, select the alias set for which you want to modify an existing binding.
3. Under Bindings, select the binding you want to modify from the list.
4. Click the Edit the selected binding button, located to the right of the Bindings list box, to display the Edit a Binding dialog box.

The top pane in the dialog box displays the current binding syntax.

Take one of the following actions:

- Enter the binding syntax directly into the top pane of the dialog box. The syntax is:

  ```
  connection : '\<logical_source_name'::'\<physical_data_source_name>.\<physical_table_name>'\'<physical_column_name>'
  ```

- In the physical tree in the bottom pane of the text box, navigate to the column in the physical data source which you want to use to generate a binding expression. Double-click the column name to add it as a binding expression to the text column.
Tip:
If you are using this method, it is recommended that you clear the text from the top pane of the dialog box before you begin.

6 When the binding is modified as you require, click OK to return to the Alias Set Manager.

Adding Bindings to an Alias Set
You can add one or more bindings to an alias set.

➤ To add a binding:
1 In the Essbase Studio Console, select Tools > Alias Set Manager.
2 Under Alias Sets, select the alias set to which you want to add bindings.
3 Complete step 3 through step 7 in “Creating Bindings for an Alias Set” on page 114 for all bindings you want to add to the selected alias set.

Deleting Bindings from an Alias Set
You can delete one or more bindings from an alias set.

➤ To delete a binding:
1 In the Essbase Studio Console, select Tools > Alias Set Manager.
2 Under Alias Sets, select the alias set from which you want to delete bindings.
3 Select the binding you want to delete and then click the Delete the selected bindings button, located to the right of the Bindings list box.

Note:
Use the Shift key to select two or more consecutive bindings from the list. Use the Ctrl key to select two or more nonconsecutive bindings from the list.

4 At the Confirm Delete prompt, click Yes.
The bindings you selected for deletion are cleared from the Bindings list.

Note:
If you do not want to be prompted to confirm deletions, select the “Do not show me this message again” check box.

Managing Alias Sets
in the Alias Set Manager, you can perform these management tasks

• Copy—see “Copying Alias Sets” on page 117
• Rename—see “Renaming Alias Sets” on page 117
• Delete—see “Deleting Alias Sets” on page 117

**Copying Alias Sets**

➤ To copy an alias set:

1. In the Essbase Studio Console, select Tools > Alias Set Manager.
2. Under Alias Sets, select the alias set that you want to copy.
3. Click the Create a copy of the selected alias set button, located to the right of the Alias Set list box.
4. At the prompt, enter a name for the alias set, and click OK.
   The copy of the alias set is displayed in the Alias Set list box.

**Renaming Alias Sets**

**Note:**

When renaming alias sets, you must follow the guidelines in Appendix C, “Naming Restrictions for Essbase Studio”.

➤ To rename an alias set:

1. In the Essbase Studio Console, select Tools > Alias Set Manager.
2. Under Alias Sets, select the alias set that you want to rename.
3. Click the Rename the selected alias set button, located to the right of the Alias Set list box.
4. At the prompt, enter a new name for the alias set, and click OK.
   The renamed alias set is displayed in the Alias Set list box

**Note:**

The renamed alias is temporarily prepended with an asterisk (*) in the Alias Sets list until you close and reopen the Alias Set Manager.

**Deleting Alias Sets**

➤ To delete an alias set:

1. In the Essbase Studio Console, select Tools > Alias Set Manager.
2. Under Alias Sets, select one or more alias sets that you want to delete.
Note:
Use the Shift key to select two or more consecutive alias sets from the list. Use the Ctrl key to select two or more nonconsecutive alias sets from the list.

3. Click the **Delete the selected alias sets** button, located to the right of the **Alias Set** list box.

4. At the **Confirm Delete** prompt, click **Yes**.

The alias sets you selected for deletion are cleared from the Alias Sets list.

Note:
If you do not want to be prompted to confirm deletions, select the “Do no show me this message again” check box.
Hierarchies Overview

Hierarchies determine how data is consolidated and navigated. For example, many businesses summarize their data monthly, roll up the monthly data to get quarterly figures, and then roll up the quarterly data to get annual figures. Some businesses may also summarize data by zip code, and then by city, state, and country. In Essbase Studio, hierarchies are the used to model dimensions in the Essbase cube. Any dimension can be used to consolidate data for reporting purposes. Levels in the dimension can come from different source database tables that are joined in either of these ways:

- By physical joins in the relational data source.
- By joins created in the minischema in Essbase Studio.

For example, a hierarchy for a MARKET dimension might look like the hierarchy shown below.

```
MARKET
   REGION
     STATE
```

In this hierarchy, REGION is a child of the hierarchy MARKET, and STATE is a child of REGION. The structure of the hierarchy carries over to the Essbase outline, where you can report on sales for individual states and consolidate state figures to report on regional sales. Note that the columns in a hierarchy could come from different tables within a relational data source; however, for text file data sources, columns in a hierarchy must come from the same text file.

When hierarchies are used in a cube schema, hierarchies determine the structure of the resulting Essbase model, providing named structures that contain:

- The hierarchical structure itself—a level-by-level sequence for consolidating data. For example, sales totals by STATE roll up to sales totals by REGION. If you had CITY as a child of STATE in your hierarchy, sales totals by CITY could roll up to sales totals by STATE.
The data filters that are placed on selected metadata elements within the hierarchy—a way to select specific categories of information. For example, you can filter the REGION column to include only information on sales in the USA.

The organizational sequence of the data—the sort sequence for a column. For example, you can sort the MONTH column in descending sequence to see the most recent totals first.

Transformations—a way to control column data values or measures, which become Essbase member names. For example, if you had CITY as a child of STATE in your hierarchy, you could ensure unique member names by prefixing each CITY value with an appropriate value from the STATE column. You could then differentiate between CA_ALBANY and NY_ALBANY.

After a hierarchy is created, it can be dragged from the logical tree directly into the Cube Schema wizard. Further, hierarchies can be reused in any number of cube schemas. By creating multiple hierarchies, you can customize a cube schema for each user group. For example, assume that you use hierarchical filters in the hierarchies that you provide to users from a specific corporate division. With such hierarchies, the users can more easily create cube schemas that contain only the data relevant to their specific division.

Essbase Studio supports standard, measure, and calendar hierarchies. See the following sections for information on creating these objects:

- “Creating Standard and Measure Hierarchies” on page 120
- “Creating Calendar Hierarchies” on page 128

Creating Standard and Measure Hierarchies

Hierarchies define the organizational structure of a group of member levels or a group of members. Use standard hierarchies to define consolidation and navigation for your business objects such as products or markets.

Use measure hierarchies to create hierarchies with dimension elements that represent your data values. If your data contains more than one measure (data value), define a measure hierarchy and add all measures to that hierarchy.

The procedure for using key bindings as columns in hierarchies is discussed in “Using Delayed Key Bindings in Hierarchies” on page 123.

Illustrations of different types of hierarchies are presented in “Hierarchy Examples” on page 125.

To create calendar hierarchies, see “Creating Calendar Hierarchies” on page 128.

To create a standard or measure hierarchy:

1. Right-click the appropriate folder in the Metadata Navigator and select New > Hierarchy or New > Measure Hierarchy to launch the hierarchy editor.

2. Enter the Name for the hierarchy.

The default name is NewHierarchy for standard hierarchies, and NewMeasureHierarchy for measure hierarchies.
3 Optionally, enter a Description.

4 Use any of the following methods, or combination of methods, to add elements to the hierarchy.

Note:

In the same hierarchy, combinations of logical elements, physical elements, and user-defined elements are possible. You may add logical elements from the Metadata Navigator and physical elements from the Source Navigator in the same hierarchy. You may also add a user-defined parent to the hierarchy and add logical or physical elements as its children.

Drag-and-drop from the Metadata Navigator:

a. In the Metadata Navigator, navigate to the location of the first metadata element you want to add to the hierarchy.

b. Select the metadata element to add, drag it to the Edit Hierarchy dialog box, and drop it under the Hierarchy column in the Data group.

c. Select the next metadata element to add to the hierarchy, drag and drop it directly on the previous element you added.

Dropping the metadata element directly atop the previous element provides the hierarchical structure.

Dropping the metadata element in the row below the previous element begins a new chain in the hierarchy.

Note:

You may also drag a physical element from the Source Navigator and drop it on top of or below the parent element. Alternatively, you can click the Add button and select “Add as child” to add a logical element to the parent element, or select “Add as sibling” to begin a new chain in the hierarchy.

d. Optional: Repeat step 4.c as many times as appropriate for the hierarchy you want to build.

Drag-and-drop from the Source Navigator:

a. In the Source Navigator, navigate to the location of the first physical element you want to add to the hierarchy.

b. Select the physical element to add, drag it to the Edit Hierarchy dialog box, and drop it under the Hierarchy column in the Data group.

For each physical element you add to the hierarchy, a corresponding metadata element is added to the Metadata Navigator in the same location.

c. Select the next physical element to add to the hierarchy, drag and drop it directly on the previous element you added.

Dropping the physical element directly atop the previous element provides the hierarchical structure.
Dropping the physical element in the row below the previous element begins a new chain in the hierarchy.

**Note:**

You may also drag a logical element from the Metadata Navigator and drop it on top of or below the parent element. Alternatively, you can click the Add button and select “Add as child” to add a logical element to the parent element, or select “Add as sibling” to begin a new chain in the hierarchy.

d. **Optional:** Repeat step 4.c as many times as appropriate for the hierarchy you want to build.

Use the Add button:

a. Click the Add button and make a selection from the pop-up menu:
   - Select **Add child** or **Add sibling**.
     
     The Select Entity dialog box is displayed. Navigate to the logical element that you want to add as a child or sibling, select it, and click OK.
     
     Note that for the first element in the hierarchy, you can choose either “Add child” or “Add sibling.”
   - Select **Add user-defined child** or **Add user-defined sibling**.
     
     Type the name of the user-defined member in the editable field in the grid under Hierarchy. The default name is “NewMemberN.”
     
     For the first element in the hierarchy, you can choose either “Add user-defined child” or “Add user-defined sibling.”

     **Note:**
     
     A user-defined element cannot be added as a child for a column-based (relational) element.

     **Note:**
     
     The Key Binding column will display the text “Unspecified”. When you save the hierarchy, the Key Binding column will be updated to show the name of the child or sibling element.

b. Click the Add button again and make another selection, as described in step 4.a.

   **Note:**
   
   If you selected the “Add child” or “Add sibling” option to create the top level of the hierarchy, then the “Add user-defined child” option is not available. The valid selections are “Add child,” “Add sibling,” or “Add user-defined sibling.”

c. **Optional:** Repeat step 4.b as many times as appropriate for the hierarchy you want to build.

5 Click **Save**.
Optionally, click “Preview” to launch the Sample Data dialog box and view the hierarchy structure, as described in “Previewing Hierarchies” on page 139. Click OK when finished previewing the hierarchy.

Note:

Hierarchy preview is not available for measure hierarchies. The members are displayed in the hierarchy itself.

Using Delayed Key Bindings in Hierarchies

The key binding in a hierarchy reflects the key binding from the metadata element. You can use a delayed key binding in a hierarchy.

You can choose to define the delayed binding or elect to have Essbase Studio generate the delayed binding. If you choose to define the key binding for a delayed element, the key binding will only be saved with that hierarchy. The key binding will not update the original metadata element.

➢ To use the key binding column in a hierarchy:

1 Create a dimension element, using the Delayed key binding option (see step 3 on page 103).

2 Right-click the appropriate folder in the Metadata Navigator and select New > Hierarchy to launch the hierarchy editor.

3 Enter the Name for the hierarchy.

The default name is NewHierarchy.

4 Optional—Enter a Description.

5 Drag-and-drop from the Metadata Navigator:

a. In the Metadata Navigator, navigate to the location of the applicable dimension element.

b. Select the dimension element, drag it to the Edit Hierarchy dialog box, and drop it under the Hierarchy column in the Data group.

The text for Key Binding will be “Delayed”.

Note:

If you save the hierarchy at this step in the procedure, the text “Delayed” in the Key Binding column will be replaced by a binding generated by Essbase Studio.

6 In the Key Binding column, click the cell for the dimension element and click the cell button.

The Edit Entity dialog box is displayed.

7 Create an expression to define the delayed binding:
**Note:**

The expression to define a delayed key binding is written in CPL (Common Platform Language). The expression is a sequence of operands and operators following the language-defined syntax. Each expression returns a value, the type of which defines the type of the expression. See Appendix D, “CPL Reference”.

a. Select the **Source** tab in the lower-left of the dialog box.
b. Expand the dimensions to display the members.
c. Select a member.
d. Drag or double-click the member to move the connection string for the member to the **Expression** text box.
e. Select the **Functions** tab.
f. Expand the SQL level to display the function types.
g. Expand the function types to display the functions.
h. Select a function.
i. Drag or double-click the function string to move it to the **Expression** text box.
j. Select the **Operators** tab.
k. Expand the operator types to display the operators.
l. Select an operator.
m. Drag or double-click the operator to move it to the **Expression** text box.

8 **Optional**—Create an expression to add filters to further refine the delayed binding definition:

a. Select the **Source** tab in the lower-left of the dialog box.
b. Expand the dimensions to display the members.
c. Select a member.
d. Drag or double-click the member to move the connection string for the member to the **Expression** text box.
e. Select the **Functions** tab.
f. Expand the SQL level to display the function types.
g. Expand the function types to display the functions.
h. Select a function.
i. Drag or double-click the function string to move it to the **Expression** text box.
j. Select the **Operators** tab.
k. Expand the operator types to display the operators.
l. Select an operator.
m. Drag or double-click the operator to move it to the **Expression** text box.

9 Click **OK**.
The Key Binding column will display the binding definition you just created.

10 Click Save.

Note:
If you save the hierarchy at this step in the procedure, the Key Binding column will become the binding definition you just created.

### Hierarchy Examples

From the basic standard and measure hierarchy types, you can create different kinds of hierarchies. See the following topics for examples

- “Single-chain Hierarchies” on page 125
- “Multi-chain Hierarchies” on page 126
- “Multi-chain Hierarchy with a Shared Member (Alternate Hierarchy)” on page 126
- “Multi-chain Hierarchy with Attribute Dimensions” on page 126
- “Recursive Hierarchies” on page 128
- “Time Hierarchies Built from DATE Type Metadata Elements” on page 128

### Single-chain Hierarchies

A single-chain hierarchy is the simplest of hierarchies. Note that the levels in a hierarchy must come from the same data source; however, the hierarchy can contain levels from both physical and logical elements.

Figure 5 is an example of a simple single-chain hierarchy for Market. This hierarchy has just two levels. The parent level is the Region column from the Region table in the TBC database. The child level is the State column from the Market table.

![Figure 5 Simple single-chain hierarchy](image)

Figure 6 is a single-chain, multi-generation hierarchy, meaning there are more than two levels in the hierarchy. Based on the FoodMart database, the levels in this hierarchy all come from the same table. Note that the levels in a multi-generation hierarchy do not have to come from the same table.
Multi-chain Hierarchies

A multi-chain hierarchy can represent alternative roll-ups for elements belonging to the same business area. Figure 7 shows a roll-up for State on Region and a roll-up for State on Country.

Multi-chain Hierarchy with a Shared Member (Alternate Hierarchy)

You can design a multi-chain hierarchy that builds dimensions with shared members, also known as alternate hierarchies.

One way to accomplish this is to first create a copy of the dimension element that you want to designate as a shared member. By making a copy, the binding expression for this element is the same as the original dimension element. Then, when you create the second hierarchy in the chain, which is the alternate hierarchy, add a user-defined member as the parent and add the new dimension element as the child, which will be the shared member, as shown in Figure 8.

Multi-chain Hierarchy with Attribute Dimensions

To produce attribute dimensions in your cube, you first build a multi-chain hierarchy, adding the same dimension element or physical column as the leaf-level member of each chain, as shown in Figure 9.
In the Cube Schema Wizard, use this hierarchy in your cube schema and build an Essbase model. After the Essbase model is built, in Essbase Model Properties, set the attribute member properties for parent of each of the attribute hierarchies in the chain. In the example in Figure 9, set the attribute property on the members Caffeinated, Ounces, PkgType, and IntroDate. See “Selecting Members as Attributes” on page 178 for more information.

To set attribute formatting rules for the model, see “Defining Attributes in Models” on page 153.

Figure 10 shows a portion of the resulting Essbase outline after cube deployment.
Recursive Hierarchies

A recursive hierarchy contains member levels that are based on the contents of the two columns of a parent-child relationship. Recursive hierarchies are generally built from recursive tables. In the TBC sample database, the Measures table is a recursive table.

Figure 11 is an example of a recursive hierarchy.

<table>
<thead>
<tr>
<th>Hierarchy</th>
<th>Key Binding</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARENT</td>
<td>tbcSource: 'tbc.measures'.PARENT</td>
</tr>
<tr>
<td>CHILD</td>
<td>tbcSource: 'tbc.measures'.CHILD</td>
</tr>
</tbody>
</table>

If a source database contains recursive tables, you need to join each recursive table to itself (self-join) to access any related information and to consolidate the data properly.

Time Hierarchies Built from DATE Type Metadata Elements

You must have metadata elements of the DATE type in order to build the time hierarchy in this example. Before you begin, be sure you have completed the procedure in “Creating Date Elements” on page 109.

Creating Calendar Hierarchies

The Edit Calendar Hierarchy dialog box provides a way to build a time dimension modeled on a standard corporate calendar. You can build a time dimension based on a standard (Gregorian) calendar, an industry specific calendar, or a customized version of one.

Note:

Varying attributes are not supported in calendar hierarchies.

For each calendar type, Essbase Studio provides a list of time period definitions from which to choose, such as year, semester, and quarter.

Essbase Studio supports hierarchies for the following calendar types:

- Gregorian—see “Gregorian Calendar Hierarchies” on page 129
- Fiscal—see “Fiscal Calendar Hierarchies” on page 129
- Retail—see “Retail Calendar Hierarchies” on page 131
- ISO—see “ISO Calendar Hierarchies” on page 132
- Manufacturing—see “Manufacturing Calendar Hierarchies” on page 132
Note:

When you use a calendar hierarchy to build a cube schema and Essbase model, you must specify a default data load binding for the lowest level in the calendar hierarchy at cube schema creation time. See “Defining Data Load Mappings” on page 140 for more information.

**Gregorian Calendar Hierarchies**

A Gregorian calendar is the standard twelve month calendar starting on Jan 01 and ending on Dec 31. Gregorian calendars can include members on year, semester, trimester, quarter, month, week, and day time periods.

To create a Gregorian calendar hierarchy:

1. Right-click the appropriate folder in the Metadata Navigator and select New > Calendar Hierarchy to launch the Edit Calendar Hierarchy dialog box.
2. In Hierarchy Name, enter the name for the Gregorian calendar hierarchy.
3. From Calendar, select Gregorian.
4. In Modeling Parameters, enter the start and end dates of the period you are modeling, and select a First day of week.
5. To define time depth, see “Time Depth” on page 134.
   
   Time depth applies to all calendar types.
6. To define day attributes, see “Day Attributes” on page 134.
   
   Day attributes apply to all calendar types.
7. To set linked value attributes, see “Linked Value Attributes” on page 135.
   
   Linked value attributes apply to all calendar types.
8. Click OK to close the Edit Calendar Hierarchy dialog box.

**Fiscal Calendar Hierarchies**

Fiscal calendar definitions are based on company reporting requirements and can start on any date. In this calendar a week has seven days. The twelve month reporting period includes two months of four weeks and one month of 5 weeks, in a repeated three-month quarterly pattern. When you select the three-month pattern of each quarter (4-4-5, 4-5-4, or 5-4-4 weeks), the months are defined based on the pattern specified. If the year has 53 weeks, one of the months can have an extra week.

The Fiscal calendar builds your selected time depths into the calendar dimension hierarchy. You can select any from the following time depths:

- Year
- Semester
- Trimester
Quarter
Month
Week
Day

To create a fiscal calendar hierarchy:

1. Right-click the appropriate folder in the Metadata Navigator and select New > Calendar Hierarchy to launch the Edit Calendar Hierarchy dialog box.

2. In Hierarchy Name, enter the name for the fiscal calendar hierarchy.

3. From Calendar, select Fiscal and click the Semantic Rules link.

4. In Year Semantic Rules and Month Semantic Rules, adjust how your fiscal year is structured by choosing from the following options.

   The selection you make in Year Semantic Rules affects the options in Month Semantic Rules.

   All options use the Month, Week, and Day drop-down boxes in this group.

Year Semantic Rules

- Starting week - number in month—Select the month and week in which your fiscal year starts from the drop down boxes.

- Ending week - number in month—Select the month and week in which your fiscal year ends from the drop down boxes.

- Year starts on a week that includes a specific date—Select the date from the month and day drop down boxes. The fiscal year begins on the first day of this week, as determined in the Select Common Attributes wizard panel.

- Year ends on a week that includes a specific date—Select the date from the month and day drop down boxes. The fiscal year ends on the last day of this week.

- Year starts on the week on or immediately following a specific date—Select the date from the month and day drop down boxes. The fiscal year begins on the first day of the week on or immediately following this date.

- Year starts on a specific date—Select the date from the month and day drop down boxes. The fiscal year begins on this date.
  - Enforce 53 weeks—The first or last week, depending on which has been chosen, will have days added to make a complete week. The days will be taken from the adjacent fiscal year.
  - Enforce 52 weeks—The first or last week, depending on which has been chosen, will be removed from the calendar and moved to the adjacent fiscal year

Month Semantic Rules

Some of these options may be inactive (grayed out) depending on the year semantic rule selections.

- Starting week number specified—The starting week number is specified by the selection in the Week drop down box above.
● Month always starts on a week that includes a specific date—The date is specified by the selection in the Day drop down box above.

● Month starts on the week on or immediately following a specific date—The date is specified by the selection in the Day drop down box above.

● By Qtr-Month pattern—The pattern of weeks per month for each quarter is specified by the drop down box. In a 53-week year, you can specify the month which includes the extra week in the Month having extra week drop down box.

● Month always starts on a specific day number—The date is specified by the selection in the Day drop down box above.

5 In Modeling Parameters, enter the start and end dates of the period you are modeling, and select a First day of week.

6 To define time depth, see “Time Depth” on page 134.

Time depth applies to all calendar types.

7 To define day attributes, see “Day Attributes” on page 134.

Day attributes apply to all calendar types.

8 To set linked value attributes, see “Linked Value Attributes” on page 135.

Linked value attributes apply to all calendar types.

9 Click OK to close the Edit Calendar Hierarchy dialog box.

Retail Calendar Hierarchies

This calendar comes from the National Retail Federation and is modeled to analyze week over week data across years. It has a 4-5-4 quarter pattern with leap weeks every 5-6 years. The starting date differs from year to year, but it always falls in early February. When comparing year over year, it is a standard practice to omit the first week of a fifty-three week year to normalize for the extra week while keeping the same set of holidays in both years. Fiscal calendars can include members on year, semester, quarter, month, week, and day time periods.

➤ To create a retail calendar hierarchy:

1 Right-click the appropriate folder in the Metadata Navigator and select New > Calendar Hierarchy to launch the Edit Calendar Hierarchy dialog box.

2 In Hierarchy Name, enter the name for the retail calendar hierarchy.

3 From Calendar, select Retail and click the Semantic Rules link.

4 In Month Semantic Rules, you can only choose By Qtr-Month Pattern and specify the following options:

● Week Pattern—The pattern of weeks per month for each quarter

● Month Having Extra Week—In a 53-week year, select the month which includes the extra week.

5 In Modeling Parameters, enter the start and end dates of the period you are modeling, and select a First day of week.
6 To define time depth, see “Time Depth” on page 134.
   Time depth applies to all calendar types.
7 To define day attributes, see “Day Attributes” on page 134.
   Day attributes apply to all calendar types.
8 To set linked value attributes, see “Linked Value Attributes” on page 135.
   Linked value attributes apply to all calendar types.
9 Click OK to close the Edit Calendar Hierarchy dialog box.

ISO Calendar Hierarchies

The ISO calendar is made up of seven day weeks. The year can start before or after the start of
the Gregorian new year (Jan 1). The year is modeled to start on a day such that the first week of
the ISO calendar contains the first Thursday of Gregorian year. The first day of the week is
defined as Monday. The ISO 8601 calendar hierarchy can only include members on year, week,
and day periods.

➢ To create an ISO calendar hierarchy:

1 Right-click the appropriate folder in the Metadata Navigator and select New > Calendar Hierarchy to
   launch the Edit Calendar Hierarchy dialog box.
2 In Hierarchy Name, enter the name for the ISO calendar hierarchy.
3 From Calendar, select ISO.
4 In Modeling Parameters, enter the start and end dates of the period you are modeling, and select a First
day of week.
5 To define time depth, see “Time Depth” on page 134.
   Time depth applies to all calendar types.
6 To define day attributes, see “Day Attributes” on page 134.
   Day attributes apply to all calendar types.
7 To set linked value attributes, see “Linked Value Attributes” on page 135.
   Linked value attributes apply to all calendar types.
8 Click OK to close the Edit Calendar Hierarchy dialog box.

Manufacturing Calendar Hierarchies

The manufacturing calendar defines a thirteen period year, made up of seven day weeks. The
periods are divided into three quarters of three periods each and one quarter of four periods.
Each period has four weeks, with the exception of one period which can have an extra week if
the year has fifty-three weeks.
When you define the thirteen periods, you specify which quarter has the extra period. If the year has 53 weeks, you must specify which period will have the extra week. If you specify that the year starts on a specific date, you must indicate whether the year has 52 weeks or 53.

The week definition determines on how to divide the calendar year into weeks. You can adjust the week definition in order to make a 52 week or 53 week year.

To create a manufacturing calendar hierarchy

1. Right-click the appropriate folder in the Metadata Navigator and select New > Calendar Hierarchy to launch the Edit Calendar Hierarchy dialog box.

2. In Hierarchy Name, enter the name for the manufacturing calendar hierarchy.

3. From Calendar, select Manufacturing and click the Semantic Rules link.

4. In Year Semantic Rules and Month Semantic Rules, adjust how your fiscal year is structured by choosing from the following options.

The selection you make in Year Semantic Rules affects the options in Month Semantic Rules.

All options use the Month, Week, and Day drop-down boxes in this group.

Year Semantic Rules

- **Starting week**—number in month: Select the month and week in which your fiscal year starts from the drop down boxes.

- **Ending week**—number in month: Select the month and week in which your fiscal year ends from the drop down boxes.

- **Year starts on a week that includes a specific date**—Select the date from the month and day drop down boxes. The fiscal year begins on the first day of this week, as determined in the Select Common Attributes wizard panel.

- **Year ends on a week that includes a specific date**—Select the date from the month and day drop down boxes. The fiscal year ends on the last day of this week.

- **Year starts on the week on or immediately following a specific date**—Select the date from the month and day drop down boxes. The fiscal year begins on the first day of the week on or immediately following this date.

- **Year starts on a specific date**—Select the date from the month and day drop down boxes. The fiscal year begins on this date.
  - Enforce 53 weeks—The first or last week, depending on which has been chosen, will have days added to make a complete week. The days will be taken from the adjacent fiscal year.
  - Enforce 52 weeks—The first or last week, depending on which has been chosen, will be removed from the calendar and moved to the adjacent fiscal year

Period Semantic Rules

Some of these options may be inactive (grayed out) depending on the year semantic rule selections.

- **Quarter having 4 periods**—Select a quarter from the drop-down box.
Period having 5 weeks—Select a period from the drop down box.

5 In Modeling Parameters, enter the start and end dates of the period you are modeling, and select a First day of week.

6 To define time depth, see “Time Depth” on page 134.
   Time depth applies to all calendar types.

7 To define day attributes, see “Day Attributes” on page 134.
   Day attributes apply to all calendar types.

8 To set linked value attributes, see “Linked Value Attributes” on page 135.
   Linked value attributes apply to all calendar types.

9 Click OK to close the Edit Calendar Hierarchy dialog box.

**Time Depth**

Time depth is the number of levels, or depth, that you want your calendar hierarchy to have.

➤ To define time depth:

1 Under Time Depth, select the check box next to a hierarchy level to include it in the hierarchy.
   Note the following:
   ● Grayed-out levels with the check box selected are required and must be used with this calendar type.
   ● Grayed-out levels with the check box cleared are not allowed with this calendar type.

2 To modify the labeling rules for a hierarchy level, expand the level and select the Edit button next to Labeling Rules.
   The Edit Labeling Rules dialog box is displayed.

3 In the Edit Labeling Rules dialog box, select a rule from the list and click OK.

**Day Attributes**

➤ To define day attributes:

1 In Edit Calendar Hierarchy, click Day Attributes.

2 In Reserved Day(s), select the check box for each day of the week desired.
   Enabling reserved days adds an attribute dimension for each day of the week selected. A default dimension name is provided, which you can overwrite in the text box under Dimension Name for the selected day of the week.
   You may specify more than one dimension name per selected day of the week. Use a comma to separate multiple dimension names. For example, if you select Sunday, and specify two dimension names, Week and Sunday, use this syntax:
Note:
The comma is a reserved character for separating dimension names in the Dimension Name text box. Dimension names cannot contain a comma as commas will be read as a separator.

3 To add a “Holiday” attribute to the dates selected, click Add and then select holidays from the calendar tool.
To remove the holiday attribute from a date, select the date and click Remove.

4 To add an attribute with the name of the day of the week to each day-level member, select Perform Day Modeling to assign day names to each day-level member.

5 Click OK to return to the Edit Calendar Hierarchy dialog box.

Linked Value Attributes
Linked value attributes (LVAs) are a type of attribute used to describe the periodicity of Time dimension members. Periodicity is any shared pattern among time dimension members that makes them meaningful for time-based analysis. For example, January and April share the periodicity of being opening months of a quarter in the Gregorian calendar.

Use this dialog box to define LVAs for those periodically recurring members that you want to aggregate for analysis. You can aggregate on parameters such as quarter by year, month by year, or week by quarter.

➤ To define LVAs, select the check box for each aggregation that you want to define, then modify the dimension name and alias prefix as necessary.

Editing Hierarchies
When you edit hierarchies, be aware that they may be used in cube schemas and Essbase models. Changing the hierarchy will cause the cube schemas and the models built it from it to be out of sync with the hierarchy. Oracle recommends recreating the cube schema and Essbase model when you make changes to the underlying hierarchies.

➤ To edit a hierarchy:

1 Select the hierarchy in the Metadata Navigator, right-click, and select Edit.

2 Perform a task:
   ● If you are editing a standard or measure hierarchy, follow the procedure in “Creating Standard and Measure Hierarchies” on page 120, starting at step 2.
   ● If you are editing a calendar hierarchy, see “Creating Calendar Hierarchies” on page 128, select the appropriate calendar type, and follow the procedure for that calendar type, starting at step 2.
Cube Schemas

In This Chapter

Cube Schema Overview ......................................................................................................... 137
Creating or Editing Cube Schemas............................................................................................. 137
Creating Essbase Models from Existing Cube Schemas............................................................ 142
Viewing Deployment History .................................................................................................... 142

Cube Schema Overview

Cube schemas are the foundation structure of Essbase cubes. In the cube schema, you select hierarchies, measures, and measure hierarchies, and you indicate any data load overrides for individual members in hierarchies. You also have the option of creating an Essbase model from the cube schema.

Creating or Editing Cube Schemas

Use the Cube Schema Wizard to create Essbase cubes and, optionally, Essbase models. You also use the same interface to edit cube schemas.

➤ To create a new cube schema, in the Logical Tree, select New > Cube Schema to launch the Cube Schema Wizard.

➤ To edit an existing cube schema, right-click the cube schema name in the Logical Tree and select Edit.

The Cube Schema Wizard consists of two default screens where you accomplish the following tasks:

● Choose measures and hierarchies—see “Choosing Measures and Hierarchies for a Cube Schema” on page 138

● Cube Schema options—see “Setting Cube Schema Options” on page 139

An optional third screen allows you to define data load bindings for selected hierarchies, measures, and measure hierarchies. See “Defining Data Load Mappings” on page 140 for more information.
Choosing Measures and Hierarchies for a Cube Schema

Choose measures and hierarchies whether you are creating editing a cube schema.

➤ To choose measures and hierarchies in the Cube Schema Wizard:

1 In Choose Measures and Hierarchies, enter a Cube Schema Name.

2 Optional—Enter an informational Comment.

3 Drag one or more hierarchies you want to include in this cube schema from the Available Dimension Elements list and drop them in the Hierarchies list on the right.

   You cannot add the same hierarchy more than once.

   Alternatively, you may add hierarchies by selecting them in the “Available Dimension Elements” list and clicking the “Add selected hierarchies” button.

   Note:

   To remove one or more hierarchies from the Hierarchies list, select the hierarchies in the Hierarchies list and click the “Remove selected hierarchies” button.

4 Drag one or more measures or a single measures hierarchy from the Available Dimension Elements list and drop them in the Measures/Measures Hierarchy list on the right.

   You cannot add the same measure or measure hierarchy more than once.

   Note:

   You may add only one measure hierarchy to the cube schema, but you may add one or more measures. You may not add both measures and a measure hierarchy to the same cube schema. You must select either a single measure hierarchy or one or more measures.

   Alternatively, you may add measures or a measure hierarchy by selecting them in the “Available Dimension Elements” list and clicking the “Add selected measures or measure hierarchy” button.

   Note:

   To remove one or more measures or a measure hierarchy from the Measures/Measure Hierarchy list, select the measures or measure hierarchy in the Measures/Measures Hierarchy list and click the “Remove selected measures or measures hierarchies” button.

5 Optional—Click Preview Hierarchies to view the structure of the hierarchies you have chosen for this cube schema, as described in “Previewing Hierarchies” on page 139.

   When you are finished viewing hierarchies, click OK.

6 Clicking Next takes you to the next page of the wizard, Cube Schema options, as described in “Setting Cube Schema Options” on page 139
**Previewing Hierarchies**

You can view a sample of the structure of a single hierarchy while creating or editing a hierarchy. You can also view the structure of one or more hierarchies while creating or editing a cube schema.

You can also preview a hierarchy at anytime from the Metadata Navigator.

To preview hierarchies:

1. **Access the Sample Data window in either of these ways:**
   - In the Choose Measures and Hierarchies page of the Cube Schema Wizard, click Preview Hierarchies.
   - Right-click a hierarchy in the Metadata Navigator and select Preview Hierarchy.
   - Select a hierarchy to view by double-clicking it in the Metadata Navigator and, in the Hierarchy editor (a tab in the work area of the Essbase Studio Console) click Preview.

   The Sample Data window is displayed.

**Note:**

When you select a Performance Management Architect hierarchy to preview, the Preview BPMA Dimension window is displayed where you will view the properties of individual members (see step 3).

2. **If previewing hierarchies from the Cube Schema Wizard, take these actions:**
   a. Click the tabs at the top of the Sample Data window to view a single hierarchy or all hierarchies in the cube schema.
   b. When finished viewing the sample, click OK to return to the first page Cube Schema Wizard.

3. **If previewing a Performance Management Architect dimension, in the Preview BPMA Dimension window, take these actions:**
   a. Expand the hierarchy in the left frame.
   b. Select a member in the hierarchy and view the properties listed in the Property Name and Property Value columns on the right.
   c. Repeat step 3.b for each member whose properties you want to view.

4. **If you are previewing a hierarchy from the Hierarchy editor or directly from the Metadata Navigator, click OK to close the Sample Data window.**

**Setting Cube Schema Options**

In the Cube Schema Wizard, use the Setting Cube Schema Options page while creating or editing a cube schema. Options you specify are whether to override the default data load mappings and to create an Essbase model from a cube schema.
To set options in the Cube Schema Wizard:

1 **Optional**—In the Cube Schema options window, select the **Override default data load bindings** check box if you want to define the default load members and data load bindings.

   **Note:**
   You must select the “Override default data load bindings” check box if your cube schema includes a calendar hierarchy.

   If the hierarchies, measures, or measure hierarchy come from different data sources, you may want to override the default data load bindings in order to minimize errors during cube deployment.

   When you select this check box, the Next button is enabled. See “Defining Data Load Mappings” on page 140 for information on completing the “Define Data Load Mappings” page of the wizard.

2 **Optional**—Select **Create Essbase Model** to create an Essbase model from this cube schema, and then complete these steps:
   a. Enter a **Model Name** for the Essbase model.
   b. If only one measure has been specified as the Measure for the cube schema, then in **Accounts Dimension**, specify the hierarchy from which the accounts dimension for this model will be generated.

      When multiple measures are specified as the Measure for the cube schema, the accounts dimension is created with the measures as the members. When measure hierarchy is specified, then it is tagged as the accounts dimension.

      To allow Essbase Studio to generate the accounts dimension, select “System generated.”

3 **Click Finish** to begin building the cube schema and, if you specified it, the Essbase model.

   Alternatively, if you selected the “Override default data load bindings” check box in step 1, click Next and follow the instructions in “Defining Data Load Mappings” on page 140.

4 **Optional**—When the process is complete, review the graphical representation of the Essbase model in the work area of the Essbase Studio Console.

---

**Defining Data Load Mappings**

Use this dialog box to override the default load bindings that will be used for loading data.

Data is loaded at the leaf level of the hierarchy; therefore, you define the data load mapping at the lowest level of the hierarchy.

You are on this page of the wizard because you selected the “Override default data load bindings” check box on the Cube Schema Options page of the wizard.
Note:

If your cube schema includes a calendar hierarchy, you must complete the procedure in this topic.

Note:

If you add data load bindings to an existing cube schema from which an Essbase model has already been created, the cube schema and model will be out of sync. If you deploy from the existing Essbase model, the model will not pick up the new data load bindings, resulting in invalid deployment results. To deploy a cube using the new data load bindings, you must create a new Essbase model and deploy from the new model.

To specify default data load bindings in the Cube Schema Wizard:

1 In the Define data load mappings window, review the hierarchies displayed in the Cube Schema Elements column to determine for which hierarchies you want to specify default data load bindings.

Note:

If your cube schema includes a calendar hierarchy, you must specify a default data load binding for the lowest level in that hierarchy. If you do not specify a default data load binding, the Essbase model cannot be built.

2 In the Data Load Binding column, click in the row of the lowest level of a selected hierarchy.

An ellipsis button, ..., displays in the row of the lowest level of each hierarchy.

3 Click ... to launch the Default Load Binding dialog box.

4 To build a binding expression, in Default Load Binding, under Formula, click the Source tab and navigate to the physical element on which you want to build a binding expression; then follow these steps:

a. Drag the physical element from Source to the Expression box.

b. Optional: Click the Functions tab and navigate to the function you want to use in your expression, then drag that function to the Expression box.

c. Optional: Click the Operators tab and navigate to the operator you want to use in your expression, then drag that operator to the Expression box.

Note:

You must place functions and operators precisely where they belong in the expression. Essbase Studio does not drop these items automatically into their correct place.

d. Optional: Manually edit the expression as necessary.

5 Optional: —Enter a Filter for the default load binding

6 Click OK to close the Default Load Binding dialog box and return to the Define data load mappings page of the Cube Schema Wizard.

7 For hierarchies that contain user-defined members, indicate the default load member as follows:
● For a pure user-defined, single-chain hierarchy, the lowest level in the hierarchy will automatically be selected for a data load, signified by a check mark in the Data Load Binding column. This check mark cannot be cleared.

● In a pure user-defined, multi-chain hierarchy, select the lowest level from only one of the chains as the default load member (signified by a check mark in the Data Load Binding column).

● For a mixed, single-chain hierarchy, where the lowest level is relational or column-based, click and complete the substeps in step 4 to build a binding expression for the default load member.

● For a mixed, multi-chain hierarchy, where the lowest level is relational or column-based, click for only one of the chains, and complete the substeps in step 4 to build a binding expression for the default load member. The default load member can be specified for only one chain in a multi-chain mixed hierarchy,

8 Click Finish to begin building the cube schema and, if you specified it, build the Essbase model.

9 Optional: When the process is complete, review the graphical representation of the Essbase model in the work area of the Essbase Studio Console.

Creating Essbase Models from Existing Cube Schemas

If you did not select the option to create an Essbase model during the cube schema creation process, you can create one at any time from the existing cube schema.

➢ To create an Essbase model from an existing cube schema:

1 In the Metadata Navigator, navigate to the cube schema from which you want to create an Essbase model.

2 Right-click the cube schema and select Create Essbase Model.

3 In the Essbase Model dialog box, enter a Model Name.

4 Optional—Provide an optional Description.

5 In Accounts Dimension, specify the hierarchy from which the accounts dimension for this model will be generated.

   To allow Essbase Studio to generate the accounts dimension, select “System generated.”

Viewing Deployment History

You can view the deployment history of a given Essbase model to see a listing of all children cubes that were deployed to various instances of Essbase Server.

➢ To view the deployment history of cubes related to an Essbase model:

1 In the Metadata Navigator, navigate to the Essbase model whose history you want to view.

2 Right-click the model name and select Show Deployment History.
A tab called “Deployment History: <Essbase Model Name>” is displayed in the work area of Essbase Studio Console, listing each deployment in a grid format.

The information shown is:

- **Status**—Provides a deployment status condition:
  - Succeeded
  - Succeeded with warnings
  - Failed

- **Time**—The date and time of the deployment.

- **Essbase Server**—The Essbase Server instance name to which the cube was deployed.

- **Application**—The Essbase application name of the cube.

- **Database**—The Essbase database name of the cube.

- **Type**—Whether the deployment was a member load, data load, or both.

- **User**—The user name of the user who deployed the cube.

3 For deployment status of **Succeeded with warnings** or **Failed**, double-click the link in the Status column for the line in the grid representing a deployment to view the *Deployment Errors and Warnings* dialog box. Any errors or warnings logged during deployment are listed here.

4 **Click OK or Cancel** to close *Deployment Errors and Warnings*.

5 **When finished viewing**, click the **X** in the tab of the *Deployment History* window to close it.
Model Properties

The following topics discuss the Essbase model properties you edit on each tab of the Essbase model properties dialog box:

- General tab—“Setting General Model Properties” on page 147
- Alias tab—“Selecting Tables as Alias Sets” on page 153
- Attributes tab—“Defining Attributes in Models” on page 153

For information on models, see “Models Overview” on page 146.

To access the Model Properties Dialog box, see “Accessing the Model Properties Dialog Box” on page 147.

About Essbase Models

An Essbase model is a logical model (star schema) that is created from tables and columns in a relational database. The Essbase model is used to generate the structure of a multidimensional database.

When you build a cube schema, you specify hierarchies, measures, and measure hierarchies to include in the cube. The Essbase model shows graphically the objects and joins that comprise a cube schema.

For detailed information on Essbase models, see “Models Overview” on page 146.

You can access the Essbase Model Properties dialog from the Metadata Navigator. See “Accessing the Model Properties Dialog Box” on page 147.
Models Overview

An Essbase model is a logical model (star schema) that is created from tables and columns in a relational database. The Essbase model is used to generate the structure of a multidimensional database.

When you build a cube schema, you specify hierarchies, measures, and measure hierarchies to include in the cube. The Essbase model shows graphically the objects and joins that comprise a cube schema.

Essbase models are based on the concept that values in a source database can be categorized as either facts or dimensions of facts. Facts are the numeric, variable values in the database, such as sales figures and the number of units sold.

Dimensions are data categories used to organize 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. Dimensions are often related to business functions. Product, Region, and Year are typical dimensions. For more information on dimensions, see “Dimensions Overview” on page 160.

Associated with facts are related data values that provide additional information, such as store locations and product IDs of units sold. An Essbase model contains a fact table, dimension tables, dimension branches, and optional time and accounts dimensions. You can also tag columns in a hierarchy to be attribute-enabled. These columns become attributes such as color or size, in the outline and provide an additional layer of reporting in Essbase.

An Essbase model that you create with Essbase Studio is a logical model, not a physical star schema. The model is a logical representation of the data values that you select from the source database tables and that you want to report in Essbase. You use a model to create outlines which contain the basic structure required to load data into Essbase.

You can create multiple models from the same schema, edit the properties of each model, and then deploy each model to one or more Essbase outlines.

Essbase models have the following features:

- They are reusable. You can use the same model as the basis for more than one outline.
- They provide a layer of abstraction that insulates the database outline from changes in the source database.
- They enable you to create hierarchies to structure and summarize the data from the source database. You can use these hierarchies in multiple outlines.
- They support OLAP and XOLAP environments.

See “OLAP Overview” on page 150 and “XOLAP Overview” on page 151.

You specify building an Essbase model either during the cube schema creation process or later, from an existing cube schema. One cube schema can have multiple Essbase models associated with it.

Use the Essbase Model Properties dialog box to modify properties of an Essbase model before you deploy it. You can also view and edit properties at other levels. See “Dimensions Overview” on page 160 and “Member Properties” on page 176.
The Essbase Model Properties dialog box has three tabs for editing model properties:

- **General tab**—“Setting General Model Properties” on page 147
- **Alias tab**—“Selecting Tables as Alias Sets” on page 153
- **Attributes tab**—“Defining Attributes in Models” on page 153

See “Accessing the Model Properties Dialog Box” on page 147.

### Accessing the Model Properties Dialog Box

➤ To access the Essbase Model Properties dialog box:

1. In the Metadata Navigator, right-click the name of the model.
2. Select **Essbase Properties**.

The General tab for models is displayed by default.

### Setting General Model Properties

Set the general properties of the Essbase model in the General tab of the Essbase Model Properties dialog box.

For information on models, see “Models Overview” on page 146.

➤ To set general properties for an Essbase model:

1. **Access the Essbase Model Properties dialog box using the procedure in “Accessing the Model Properties Dialog Box” on page 147.**

   The model and cube schema names are displayed in the read-only Name and Cube Schema fields.

2. **Optional:** Enter an explanatory note in the **Comment** text box, for example, “To be used in conjunction with April analysis.”

3. **Optional:** The **Last deployed** field displays information about the most recent deployment. Click the button to right of the field to view the complete deployment history of the model.

   **Note:**
   
   If the current model has never been deployed, the Last deployed field is blank.

4. **Optional: Select one or more of the following options:**

   - **ASO storage model**—Enables the model to support aggregate storage.

     Select this option to store data in aggregate storage format. Use this option when the Essbase outline contains a large number of sparse dimensions and one or no dense dimensions.

     If you select the ASO storage model option, the following message will be displayed:
“Setting a model to aggregate storage can sometimes require adjustments to some of the settings. Click ‘Yes’ if you want Essbase Studio to inspect the model and make these adjustments to the model, otherwise, click ‘No’.”

The intent of these adjustments is to eliminate potential validation warnings and errors, and it is recommended that you select “Yes.” If you select “No.” you can continue all operations with the model, but you are less certain of avoiding validation warnings and errors.

- **Duplicate member name support**—Enables the model to support duplicate member names.

  In an Essbase outline that supports duplicate member names, duplicates are allowed in different dimensions and under different parent levels. This greatly enhances ease-of-use in creating analytic applications where members (such as cities in a customer dimension and a supplier dimension) may have names duplicated.

- **Use Unicode character set**—Enables the model to support Unicode character sets.

  Use this option if you want your Essbase application to encode character text in UTF-8, enabling users with computers set up for different languages to share application data.

- **XOLAP Model**—Enables the model to function as an XOLAP model.

  See “XOLAP Overview” on page 151.

5 **Optional:** Click the **Custom data load settings** button to access the **Define Dataload Settings** dialog box where you can customize SQL for data loads.

Note:

This dialog box is enabled only when all the elements in a model are sourced from relational sources. If you do not customize SQL for data loads, the message “Custom SQL enabled: false” is displayed to the right of the “Custom data load settings” button.

See “Overriding Standard Data Load SQL” on page 148 for more information.

6 Click **Apply**.

Note:

To restore the settings on this tab to their previously saved values, click Restore. To restore the settings to their original system default values, click Default.

**Overriding Standard Data Load SQL**

Use this dialog box to edit the standard SQL statements generated by Essbase Studio for use during the data load process.

Tip:

Review the “Data Load SQL Override Editing Guidelines” on page 149 for tips on editing the standard data load SQL.
To override the standard data load SQL statements:

1. Access the Essbase Model Properties dialog box using the procedure in “Accessing the Model Properties Dialog Box” on page 147.
2. Click the Custom data load settings button to access the Define Dataload Settings dialog box.
3. Select Use Custom SQL for data load.
4. By default, the SQL statements in the left text box are displayed as a list. To see the SQL statements as a folder, select the View statements as folder option.
5. In the left text box, highlight the SQL statement you want to modify.

   Note:
   It is not necessary to highlight the entire SQL statement. When you highlight a portion of an SQL statement to copy and paste to the right text box (see the next step, below), the entire statement will be copied and pasted.
6. Use the direction button to paste a copy of the SQL statement in the right text box.
7. Edit the SQL statement.

   Note:
   As you modify SQL statements, the total number of statements modified is displayed in the “Modified statements” message.
8. Click OK.

**Data Load SQL Override Editing Guidelines**

You can edit the standard SQL generated by Essbase Studio Server for use during data loads. Your edited SQL can be selected to improve performance when loading data into an Essbase database.

When you edit the standard SQL generated by the Essbase Studio Server, use the following guidelines:

- The order of columns in the SELECT clause of the edited SQL should match the order of columns in the SELECT clause of the standard SQL.
- User-defined members in the dimension tagged Accounts must be listed in the SELECT clause of the user-defined SQL.
- The number of edited SQL statements need not match the number of standard SQL statements.
- The default number of columns created as a result of a data load is one column per dimension plus all additional data columns. If a member is prefixed with previous members of its dimension (for example, parent or all ancestors), more columns are returned.
- If some columns in the data load SQL statements are NULL, you can add SQL statements to load the data at the next level in the outline. This is known as NULL Promotions.
Because data load SQL statements are tied to filter sets, you can write multiple sets of SQL statements, one for each filter set.

Data load SQL statements are associated with an outline.

**Note:**

Note: The SQL table alias generation process skips reserved keywords for SQL. Be sure to verify that your user-defined SQL is not using reserved keywords.

**OLAP Overview**

OLAP (online analytical processing) is a multidimensional, multiuser, client-server computing environment. It is intended for users who analyze consolidated enterprise data in real time. OLAP systems feature drill-down, data pivoting, complex calculations, trend analysis, and modeling.

OLAP is designed for business managers who need to address complex “what if” questions by creating scenarios to test planning strategies. Users can analyze relationships between data categories such as:

- How did Product A sell last month? How does this compare to the same month over the last five years?
- Did commissions and pricing affect how salespeople sold Product A?
- How will Product B sell next month?
- Did Product B sell better in particular regions?
- Did customers return Product C last year? Were returns due to defects? Did a specific plant manufacture defective products?

You can use Essbase Studio to build a multidimensional Essbase database to answer these types of questions quickly.

**Multidimensional Databases**

A multidimensional database (MDDB) stores consolidated data at the intersections of its members and dimensions. For example, if a company sells 20 units of products in the East region in the first quarter, Essbase stores 20 at the intersection of Product, East, Quarter1, and Unit Sales.

In a multidimensional database, a dimension is a data category representing a core component of a business plan, and it often relates to a business function. Product, Region, and Year are typical dimensions. In most databases, dimensions rarely change over the life of the application.

In a multidimensional database, a member is an individual component of a dimension. For example, Product A and Product B are members of the Product dimension. Each member has a unique name. A dimension can contain many members. In some dimensions, members change frequently over the life of the application.

Members can be parents of some members and children of others. The Essbase outline indents members below one another to indicate a consolidation relationship.
XOLAP Overview

XOLAP (extended online analytic processing) is a variation on the role of OLAP in business intelligence. Specifically, XOLAP is an Essbase multidimensional database that stores only the outline metadata and retrieves data from a relational database at query time. XOLAP thus integrates a source relational database with an Essbase database, leveraging the scalability of the relational database with the more sophisticated analytic capabilities of a multidimensional database. Your business needs determine whether OLAP or XOLAP is better suited to your environment.

For information on OLAP, see “OLAP Overview” on page 150.

OLAP and XOLAP store the metadata outline and the underlying data in different locations:
- In OLAP, the metadata and the underlying data are located in the Essbase database.
- In XOLAP, the metadata is located in the Essbase database and the underlying data remains in your source relational database.

The differences in the locations of the metadata and data are key to understanding how XOLAP can be of benefit as you use Essbase Studio because these differences affect the functionality of OLAP and XOLAP.

OLAP lends itself to traditional relational data storage and data analysis. XOLAP lends itself to operations supported in mixed or “hybrid” environments such as Hybrid Analysis and Advanced Relational Access (familiar to users of Essbase and Essbase Integration ServicesOracle Essbase Integration Services). Many of the basic concepts of Hybrid Analysis and Advanced Relational Access have been folded into the functionality of XOLAP cubes in Essbase Studio.

XOLAP Workflow

The workflow of data retrieval in an XOLAP environment is much like that of a non-XOLAP environment:

1. The model is designated as XOLAP-enabled.
   See “Designating a Model for XOLAP” on page 152.
2. The cube is deployed; however, no data is loaded at that time.
3. The Essbase database is queried, using Smart View, Oracle Essbase Visual Explorer, or another reporting tool that can access an Essbase database.
4. Essbase dynamically generates the required SQL to retrieve the data from the source relational database.

Guidelines for Using XOLAP

XOLAP has several restrictions and several unsupported usages.

Restrictions For XOLAP

XOLAP has the following restrictions:
No editing of an XOLAP cube is allowed. To modify an outline, you must create a new outline in Essbase Studio. XOLAP operations will not automatically incorporate changes in the structures and the contents of the dimension tables after an outline is created.

When derived text measures are used in cube schemas to build an Essbase model, XOLAP is not available for the model.

XOLAP can be used only with aggregate storage. The database is automatically duplicate-member enabled.

Alternate hierarchies and attribute dimensions are supported; however, attribute hierarchies are not supported.

XOLAP supports dimensions that do not have a corresponding schema-mapping in the catalog; however, in such dimensions, only one member can be a stored member.

A model that is designated as XOLAP-enabled must be deployed to a new Essbase database because incremental builds for XOLAP are not supported.

**Usages Not Supported in XOLAP**

XOLAP does not support the following usages:

- Flat files
- Ragged hierarchies
- Alternate hierarchies
- Recursive hierarchies
- Calendar hierarchies
- Filters
- Text measures
- User-defined members at the leaf level
- Multiple relational data sources

**Designating a Model for XOLAP**

You can designate a model to be enabled for XOLAP.

For information on models, see “Models Overview” on page 146.

For information on XOLAP, see “XOLAP Overview” on page 151.

To designate a model for XOLAP:

1. **Access the Essbase Model Properties dialog box using the procedure in “Accessing the Model Properties Dialog Box” on page 147.**

   The model and cube schema names are displayed in the read-only Name and Cube Schema fields.

2. **Select the check box XOLAP Model.**
To select tables as alias sets for an Essbase model:

1. Access the **Essbase Model Properties** dialog box using the procedure in “Accessing the Model Properties Dialog Box” on page 147.
2. Select the **Alias** tab.
3. Select the name of a table in **Available Tables** or in **Selected Tables**.
4. Use the direction buttons to move the table from **Available Tables** to **Selected Tables** or from **Selected Tables** to **Available Tables**.

Note:
You can also double-click on a table to move it.

5. **Click Apply.**

Note:
To restore the settings on this tab to their previously saved values, click Restore. To restore the settings to their original system default values, click Default.

### Defining Attributes in Models

In the Attributes tab of the Essbase Model Properties dialog box, you define the way that attribute dimensions are created in the model.

Use attributes to retrieve and analyze data in terms of characteristics, or attributes, of dimensions. For example, you can analyze product profitability based on size or packaging, and you can make more effective conclusions by incorporating market attributes, such as the population size of each market region, into your analysis.
For information on attributes, see “Attributes Overview” on page 154.

For information on varying attributes, see “Varying Attributes Overview” on page 155.

The following topics discuss the Essbase model properties that you edit in the Attributes tab:

- “Setting Attribute Member Names Format” on page 156
- “Setting Attribute Calculations Member Names Format” on page 157
- “Specifying Attribute Boolean, Date, and Numeric Ranges” on page 158

**Attributes Overview**

Attributes describe characteristics of data such as product size and color. Through attributes, you can group and analyze members of dimensions based on their characteristics.

Attribute analysis can tell you, for example, that decaffeinated drinks sold in cans in small markets are less profitable than you had anticipated. For more details, you can filter your analysis by specific attribute criteria, including minimum or maximum sales and profits of different products in similar market segments.

You can select, aggregate, and report on data based on common features, and you can choose from several consolidation methods:

- Sums
- Counts
- Averages
- Minimums
- Maximums

There are several attribute types:

- Text
- Numeric
- Boolean
- Date

As the following examples illustrate, analysis-by-attribute can provide depth and perspective, helping you make better-informed decisions:

- You can select, aggregate, and report on data based on common features (attributes).
- By defining attributes as having a text, numeric, Boolean, or date type, you can filter (select) data using type-related functions such as AND, OR, NOT, <, >, and = comparisons.
- You can use the numeric attribute type to group statistical values by attribute ranges; for example, population groupings such as <500,000, 500,000–1,000,000, and >1,000,000.
- You can view sums, counts, minimum or maximum values, and average values of attribute data.
- You can perform calculations using numeric attribute values in calc scripts and member formulas.
You can drill down through data to find out more detailed information, or drill up to see a summary overview of data.

**Varying Attributes Overview**

A product typically has attributes that describe or define the product. For example, a product could have an attribute describing the size of the product in ounces and an attribute describing the flavor of the product. In such a scenario, Product would be a base dimension while Ounces and Flavor would be attribute dimensions.

**Note:**

For a full explanation of base dimensions and attribute dimensions, see the Essbase *Database Administrator’s Guide*.

A varying attribute enables you to track two values in relation to a third dimension called an independent dimension. You could, for example, track your product in eight ounces over a year. In this scenario, Time is the independent dimension. The value of this third factor can vary (hence the name). For example, you could track your product over a year, a quarter, or a month.

As another example, consider this scenario: The sales representative for a client changes in midyear. Customer sales totals and sales representative assignments over six months are as follows:

| Table 3  Varying Attribute Example: Salesperson Changing Over Time |
|-------------------------|-----------------|-----------------|-----------------|----------------|-----------------|
| March | April | May | June | July | August |
| 4000 | 6000 | 2000 | 1000 | 1000 | 7000 |
| Jones | Jones | Jones | Smith | Smith | Smith |

In this example, Sales Representative is the varying attribute. Data retrievals show that the sales representative Jones sold the customer a total of $12,000 worth of products from March through May and the sales representative Smith then sold a total of $9,000 worth of products to the customer from June through August. Without using the varying attribute, the only known sales representative would be the current representative Smith to whom all sales ($21,000) would be credited.

Varying attributes offer alternate ways of grouping your members. For example, you can use color to group SKUs. In this scenario, the attribute dimension “Color” is associated with SUBSKU:

```
Product_H
  |__Family
  |   |
  |__SKU
  |   |
  |__SUBSKU
```

Model Properties  155
When Color is set as a varying attribute in Essbase Studio, the retrieval results would be similar to the following table:

<table>
<thead>
<tr>
<th>SUBSKU</th>
<th>SKU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red</td>
<td>100</td>
</tr>
<tr>
<td>White</td>
<td>400</td>
</tr>
<tr>
<td>White</td>
<td>600</td>
</tr>
<tr>
<td>Black</td>
<td>200</td>
</tr>
<tr>
<td>Black</td>
<td>300</td>
</tr>
<tr>
<td>Silver</td>
<td>500</td>
</tr>
</tbody>
</table>

Varying attributes must have multiple chains, and the leaf levels must match:

**Setting Attribute Member Names Format**

You can ensure that an attribute name is unique by assigning a prefix or suffix. Note that an outline does not display the full attribute member name after you have assigned a prefix or suffix identifier. You can view the full attribute names when you retrieve information; for example, full attribute names are displayed when you view a spreadsheet.

To set the format for attribute member names in an Essbase model:

1. Access the Essbase Model Properties dialog box using the procedure in "Accessing the Model Properties Dialog Box" on page 147.
2. Select the Attributes tab.
3. Find the Member name format grouping at the top of the dialog box.
4. Select a Value to be used as a prefix or suffix:

   - None
   - Parent
   - Grandparent
   - Ancestor

   **Note:**
   You can attach both a prefix and a suffix to a member name.
Select a Separator to be used with the prefix or suffix:
- Underscore
- Pipe
- Caret

Select a Prefix or suffix to define whether the selected value is to be used as a prefix or a suffix.
The Sample field displays an example attribute name based on the selections you have chosen.

Click Apply.

Note:
To restore the settings on this tab to their previously saved values, click Restore. To restore the settings to their original system default values, click Default.

Setting Attribute Calculations Member Names Format
You can set the format for attribute calculation member names, but an attribute calculation member name cannot be the same as a member name.

For information on attribute calculations, see “Attribute Calculations Overview” on page 158

To set the format for attribute calculations member names in an Essbase model:

1 Access the Essbase Model Properties dialog box using the procedure in “Accessing the Model Properties Dialog Box” on page 147.

2 Select the Attributes tab.

3 Find the Attribute calculations member names grouping in the center of the dialog box.

4 Optional: In the Dimension text box, rename the Attribute Calculations Dimension.

5 Optional: In the Sum member text box, rename the Sum member. For example, Sum can be renamed Total.

This attribute sums members based on their consolidation property or formula. Sum calculates a sum, but it does not always calculate a simple sum. Sum totals members based on their consolidation property or formula. For example, you can Sum the Profit% for 12-Ounce drinks, and the Sum will be based on the Profit% formula, which is profit divided by sales, with the resulting ratio multiplied by 100.

6 Optional: In the Count member text box, rename the Count member.

This attribute calculates the number of members with the specified attribute or combination of attributes. Count includes only those members that have data blocks in existence.

7 Optional: In the Minimum member text box, rename the Minimum member.

This attribute calculates the minimum data value for a specified attribute or combination of attributes.

8 Optional: In the Maximum member text box, rename the Maximum member.
This attribute calculates the maximum data value for a specified attribute or combination of attributes.

9 **Optional:** In the **Average member** text box, rename the **Average member**.

This attribute calculates a mechanical mean (average) of the values for a specified attribute or combination of attributes. The average includes only those members that have data blocks in existence (sum divided by count).

10 **Click Apply.**

**Note:**

To restore the settings on this tab to their previously saved values, click **Restore**. To restore the settings to their original system default values, click **Default**.

---

**Attribute Calculations Overview**

Attribute calculations have the following properties:

- **System-defined:** When you create an attribute dimension in an outline, the resulting reports and spreadsheets have five predefined calculations available for all attribute members.
- **Dynamic Calc:** All attribute calculations are calculated when a user requests the calculation, and then the calculation is discarded. You cannot store the calculated data in a database.
- **Displayed in Reports:** Attribute calculation results are not displayed in the outline. The calculation results are available only in spreadsheets and reports.

---

**Specifying Attribute Boolean, Date, and Numeric Ranges**

You can specify the ranges of Boolean, date, and numeric values for a member.

To specify attribute Boolean, date, and numeric ranges for a member:

1. **Access the Essbase Model Properties** dialog box using the procedure in “Accessing the Model Properties Dialog Box” on page 147.
2. **Select the Attributes** tab.
3. **Find the Boolean, date, and numeric ranges grouping near the lower-center of the dialog box.**
4. **In the True members text box, enter an attribute member name.**
   - If the member name True is used elsewhere in the outline, enter Yes, Allowed or another appropriate name in the True Member Name text box. The default member name is True.
5. **In the Dates are drop-down list, select the date format.**
   - **mm-dd-yyyy**—For example, October 19, 2000 is displayed as 10-19-2000.

**Note:**

For date type attribute members, a default mm-dd-yyyy format is assigned unless you specify a different value.
• dd-mm-yyyy—For example, October 19, 2000 is displayed as 19-10-2000.

Note:

If you change the date member name format, the names of existing members may become invalid. For example, if the 10-19-2000 member exists, and you change the format to dd-mm-yyyy, this member may become invalid.

6 In the False members text box, enter an attribute member name.

If the member name False is used elsewhere in the outline, enter Yes, Allowed or another appropriate name in the False Member Name text box. The default member name is False.

7 In the Numerics are drop-down list, specify the numeric range:

• Selecting Tops of ranges sets the names of numeric attributes to the value at the top of the numeric range. If the range is 3,000,001 to 6,000,000, selecting this option sets the name as 6,000,000.

• Selecting Bottoms of ranges sets the name of numeric attributes to the value at the bottom of the numeric range. If the range is 3,000,000 to 5,999,999, selecting this option sets the name as 3,000,000.

8 Click Apply.

Note:

To restore the settings on this tab to their previously saved values, click Restore. To restore the settings to their original system default values, click Default.

Dimension Properties

The following topics discuss the dimension properties you edit in each tab of the Essbase model properties dialog box at the dimension level:

• General tab—“Setting General Dimension Properties” on page 162
• Info tab—“Editing Dimensions” on page 163
• Account Info tab—“Editing Account Dimensions Properties” on page 168
• Formula tab—“Adding Formulas to Dimensions” on page 171
• Alias tab—“Displaying Dimension Aliases” on page 172
• UDAs tab—“Assigning User-Defined Attributes to Dimensions” on page 173
• Outline Build tab—“Selecting Outline Build Options” on page 174

For information on dimensions, see “Dimensions Overview” on page 160.

To access the dimension properties tabs, see “Accessing the Dimension Properties Tabs” on page 161.
Dimensions Overview

Dimensions are data categories used to organize 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.

Dimensions represent the core components of a business plan and often relate to business functions. Product, Region, and Year are typical dimensions. In most databases, dimensions are static, rarely changing over the life of the application.

A member is an individual component of a dimension. For example, Product A, Product B, and Product C might be members of the Product dimension. Each member has a unique name. A dimension can contain an unlimited number of members.

The following topics are discussed in this overview:

- “Dimension Types” on page 160
- “Rules for Using Time Dimensions” on page 161
- “Rules for Using Accounts Dimensions” on page 161

To access the dimension properties tabs, see “Accessing the Dimension Properties Tabs” on page 161.

Dimension Types

When you tag a dimension as a specific type, the dimension can access built-in functionality designed for that type. For example, if you define a dimension as type accounts, you can specify accounting operations for the member levels of the dimension.

There are three types of dimensions:

- Standard—Has no special type or functionality.

  Note:
  
  If you do not specify a type when you create a dimension, the dimension will be created as a standard dimension.

- Time—Describes how often you collect and update data. The dimension tagged as time enables several accounts dimension functions, such as first time balance and last time balance.

  For guidelines on using time dimensions, see “Rules for Using Time Dimensions” on page 161.

- Accounts—Contains items that you want to measure, such as profit or inventory. The dimension tagged as measure in an outline corresponds to the dimension tagged as accounts in the related Essbase database.

  Note:
  
  The accounts dimension is selected during the creation of an Essbase model. Only one dimension in the model can be the accounts dimension.
For guidelines on using accounts dimensions, see “Rules for Using Accounts Dimensions” on page 161.

**Rules for Using Time Dimensions**

Follow these rules when tagging a time dimension:

- You can tag only one dimension as time.
- When you tag a dimension as time, all members in the dimension inherit the time property.
- You can create multiple hierarchies inside the dimension tagged as time; for example, you can specify a Year, Quarter, Month hierarchy and a Year, Season hierarchy in the same time dimension.
- You can create an outline that does not have a dimension tagged as time.
- You can add time members to a dimension that is not tagged as time.
- The time dimension and the accounts dimensions are calculated before other dimensions in the database.
- You can calculate members of the time dimension on a second pass through the outline. For an overview of two pass calculation, see “Two Pass Calculation Overview” on page 166.

**Rules for Using Accounts Dimensions**

Follow these rules when tagging an accounts dimension:

- You can tag only one dimension as accounts.
- When you tag a dimension as accounts, all members in the dimension inherit the accounts property.
- You can create an outline that does not have a dimension tagged as accounts. In this scenario, an accounts dimension containing one measure will be created for you.
- The time dimension and the accounts dimensions are calculated before other dimensions in the database.
- You can calculate members of the accounts dimension on a second pass through the outline. For an overview of two pass calculation, see “Two Pass Calculation Overview” on page 166.

**Accessing the Dimension Properties Tabs**

To access the dimension properties tabs of the *Essbase Model Properties* dialog box:

1. In the *Metadata Navigator*, right-click the name of the model.
2. Select *Essbase Properties*.
3. Expand the model to display the dimension names.
4 Select a dimension.

The General tab for dimensions is displayed by default.

**Setting General Dimension Properties**

In the General tab, you can set general properties for a dimension such as defining a user-friendly name, adding comments, and specifying member format strings.

For information on dimensions, see “Dimensions Overview” on page 160.

You can set general properties for a dimension such as defining a user-friendly name, adding comments, and specifying member format strings.

➤ To set general properties for a dimension in an Essbase model:

1. Access the dimension properties tabs of the **Essbase Model Properties** dialog box using the procedure in “Accessing the Dimension Properties Tabs” on page 161.

   The Name text box displays the current name of the selected dimension.

2. Optional: Enter a user-friendly name for the dimension in the cube in the **Name in Cube** text box.

   The Source element path displays the path of the hierarchy element in which the currently selected dimension is located.

   **Note:**

   Wildcard characters such as <, [, [, *, and others cannot be used in cube names. See “Naming Restrictions for Metadata Elements” on page 272.

3. Optional: Add a comment or select a comment.

   A comment for a Market dimension might be: “Continental US, AK, and HI.” If comments exist in the database column, the Comments field will display a drop-down list of those comments.

   **Note:**

   The drop-down list box is not enabled for hierarchy elements. For members of the Accounts dimension, the drop-down list box is not available when the Accounts dimension was created from the fact table.

4. Click **Apply**.

   **Note:**

   To restore the settings on this tab to their previously saved values, click Restore. To restore the settings to their original system default values, click Default.
Editing Dimensions

In the Info tab, you can edit a dimension so that it is used in the appropriate manner in your environment. You can define the dimension type to take advantage of its built-in functionality, and you can select the best storage methods for the dimension and its data. You can also define calculations and select a sort order.

For information on dimensions, see “Dimensions Overview” on page 160.

The following topics describe the dimension properties you edit in the Info tab:

- “Selecting the Dimension Type” on page 163
- “Using Dynamic Time Series” on page 164
- “Selecting the Dimension Storage Method” on page 165
- “Selecting Two Pass Calculation” on page 165
- “Selecting the Data Storage Method” on page 166
- “Selecting the Solve Order” on page 167

Selecting the Dimension Type

When you tag a dimension as a specific type, the dimension can use the built-in functionality designed for that type.

For information on dimension types, see “Dimensions Overview” on page 160.

➤ To select a dimension type:


2. Select the Info tab.

3. Find the Dimension Type grouping in the top–left of the dialog box.

4. Select a Dimension Type:

- **Standard**—Tags the dimension as not being any special type.
- **Accounts**—Tags the dimension as an accounts dimension. This dimension type contains items that you want to measure, such as profit or inventory.
  
  For guidelines on tagging an accounts dimension, see “Rules for Using Accounts Dimensions” on page 161.

  **Note:**
  
  In some models, the accounts dimension is determined when the model is created and cannot be changed. In such a case, the Accounts option is disabled.

- **Time**—Tags the dimension as a time dimension. This dimension type describes how often you collect and update data. The dimension tagged as time enables several accounts dimension functions, such as first time balance and last time balance.
For guidelines on tagging a time dimension, see “Rules for Using Time Dimensions” on page 161.

If you tag a dimension as a time dimension, the Dynamic series button is enabled. Selecting this button opens the Dynamic Time Series dialog box which enables period-to-date reporting in block storage. See “Using Dynamic Time Series” on page 164.

5 Click Apply.

Note:

To restore the settings on this tab to their previously saved values, click Restore. To restore the settings to their original system default values, click Default.

Using Dynamic Time Series

Use the Dynamic Time Series dialog box to enable and disable Dynamic Time Series members and to associate them with a generation. You can use eight predefined Dynamic Time Series members to calculate dynamically period-to-date values. You can also specify alias names for Dynamic Time Series members.

➢ To set the values of a dynamic time series:

1 Access the dimension properties tabs of the Essbase Model Properties dialog box using the procedure in “Accessing the Dimension Properties Tabs” on page 161.

2 Select the Info tab.

3 Find the Dimension Type grouping in the top-left of the dialog box.

4 Select Time as the Dimension Type.

5 Select the Dynamic series button.

6 Select one of the following predefined Dynamic Time Series by selecting the check box to the left of the applicable series:

● H-T-D: History-to-date
● Q-T-D: Quarter-to-date
● Y-T-D: Year-to-date
● S-T-D: Season-to-date
● P-T-D: Period-to-date
● M-T-D: Month-to-date
● W-T-D: Week-to-date
● D-T-D: Date-to-date

7 Click in the Generation column, and select a generation level from the drop-down list.

8 Enter one or more aliases for the time series member.
The column of alias sets is displayed only when alias sets have been added to the model (see “Creating Alias Sets” on page 114).

Note:
If you enter an alias for the dynamic time series member and later remove the alias set from the model, the alias information you entered here for dynamic time series will be lost.

9 Click OK.
10 On the Info Tab, click Apply.

Selecting the Dimension Storage Method

The storage method you select for your dimension is dependent upon whether it is dense or sparse.

➤ To select a dimension storage method:

1 Access the dimension properties tabs of the Essbase Model Properties dialog box using the procedure in “Accessing the Dimension Properties Tabs” on page 161.
2 Select the Info tab.
3 Find the Dimension Storage grouping in the left-center of the dialog box.
4 Select a Dimension Storage:
   ● Dense—Use dense storage for a dimension in which a high percentage of available data positions are filled. For example, time data often exists for almost all products in all markets, so Year is frequently a dense dimension.
   ● Sparse—Use sparse storage for a dimension in which a low percentage of available data positions is filled. For example, if Product represents product units and Market represents geographical regions in which products are sold, then the Product and Market dimensions may be sparse dimensions because not every product is usually sold in every market.
   ● Existing—Use the storage method currently set for the dimension.
5 Click Apply.

Note:
To restore the settings on this tab to their previously saved values, click Restore. To restore the settings to their original system default values, click Default.

Selecting Two Pass Calculation

Two Pass Calculations are needed when the value of a child depends upon the value of the parent or the value of another member.

For an overview of Two Pass Calculations, see “Two Pass Calculation Overview” on page 166.
To select Two Pass Calculation:

2. Select the Info tab.
3. Find the Two Pass Calculation grouping in the top-right of the dialog box.
4. Select Two Pass Calculation.
5. Click Apply.

Note:
To restore the settings on this tab to their previously saved values, click Restore. To restore the settings to their original system default values, click Default.

Two Pass Calculation Overview

By default, Essbase outlines are calculated from the bottom up: First the values for the children are calculated and then the value for the parent is calculated. Sometimes, however, the value of a child depends upon the value of the parent or the value of another member.

To obtain the correct values for dependent dimensions and members, the outline is first calculated, and then the dimensions and members that are dependent on the calculated values of other dimensions and members are recalculated. Dimensions and members that are calculated on a second pass through the Essbase outline are called two pass calculations.

For example, to calculate the ratio between Sales and Margin, Essbase must calculate Margin (a parent member based on its children, including Sales). To ensure that the ratio calculation is based on a newly calculated Margin figure, tag the Margin % ratio member as a two pass calculation. Essbase calculates the database and then recalculates the Margin % member. The second calculation produces the correct result.

Note:
Although two pass calculation is a property that you can give to any member, it works only on members of accounts dimensions, dynamic calculation members, and dynamic calculation and store members. If you assign two pass calculation to other members, Essbase ignores it.

Selecting the Data Storage Method

You can determine how and when Essbase stores data values for a dimension. For example, you can tell Essbase to calculate the value for a dimension only when a user requests it and then to discard the value.

To select the data storage method for a dimension:

2 Select the Info tab.

3 Find the Data Storage grouping in the right-center of the dialog box.

4 Select a Data Storage method:
   - **Store Data**—Stores the data value with the member. This is the default.
   - **Dynamic Calc and Store**—Calculates the data value when a user requests it and then stores the data value.
   - **Dynamic Calc**—Calculates the data value when a user requests it and then discards the data value.

   **Note:**
   When the Dynamic Calc and Store setting or the Dynamic Calc setting is used with an aggregate storage outline, warnings may be generated during outline validation and data loads. These warnings are usually status messages and may be ignored unless your cube deployment is not successful.

   - **Never Share**—Does not allow members to be shared implicitly.
   - **Label Only**—Creates a member that is used for navigation. A label-only member contains no data value.

   **Note:**
   If you are using an aggregate storage outline, you should be aware of guidelines for selecting the Label Only option. See “Aggregate Storage with Label Only Option” on page 186.

   - **Existing**—Use the existing data storage method.

5 Click Apply.

   **Note:**
   To restore the settings on this tab to their previously saved values, click Restore. To restore the settings to their original system default values, click Default.

   **Note:**
   When the Dynamic Calc and Store setting or the Dynamic Calc setting is used with an aggregate storage outline, warnings may be generated during outline validation and data loads. These warnings are usually status messages and may be ignored unless your cube deployment is not successful.

**Selecting the Solve Order**

The Solve Order option allows you to specify a number to represent the order in which dimensions or members are calculated.

For information on using the Solve Order option, see “Solve Order Overview” on page 184.
To select a solve order:

2. Select the Info tab.
3. Find the Solve Order grouping in the lower-center of the dialog box.
4. Select a Solve Order option:
   - Dimension Solve Order—Specify a number between 1 and 127 to represent the order in which all members of the dimension are calculated.
     
     Note:
     
     The default dimension solve order is –1 which signifies that the current solve order in the Essbase cube will be used.
   - Member Solve Order—Specify a number between 1 and 127 to represent the order in which the dimension level is calculated.
     
     Note:
     
     The default member solve order is –1 which signifies that the current solve order in the Essbase cube will be used.
5. Click Apply.

Note:
To restore the settings on this tab to their previously saved values, click Restore. To restore the settings to their original system default values, click Default.

**Editing Account Dimensions Properties**

In the Account Info Tab, you can edit the properties of accounts dimensions to select optimal time balancing, instruct Essbase how to process missing values or zeroes, and select a method for reporting variances between actual and budget data.

Note:
The properties in the Dimension Properties—Account Info Tab are applicable only to accounts dimensions.

For information on dimensions, see “Dimensions Overview” on page 160.

The following topics describe the dimension properties that you edit in the Account Info tab:

- “Selecting a Time Balance” on page 169
- “Selecting the Variance Reporting Method” on page 170
- “Selecting the Skip Option” on page 170
**Selecting a Time Balance**

You can select a time balance method to determine the calculation method of parent members in a time dimension.

For information on how time balances function, see “Time Balance Overview” on page 169.

To select a time balance:

2. Select the accounts dimension.
3. Select the Account Info tab.
4. Find the Time Balance grouping in the top-left of the dialog box.
5. Select a Time Balance option:
   - None — The default value. When you set the time balance property as none, Essbase rolls up parents in the time dimension in the usual way—the value of a parent is based on the formulas and consolidation properties of the children of the parent.
   - First — Set the time balance as first when you want the parent value to represent the value of the first member in the branch (often at the beginning of a time period).
   - Last — Set the time balance as last when you want the parent value to represent the value of the last member in the branch (often at the end of a time period).
   - Average — Set the time balance as average when you want the parent value to represent the average of the children values.
   - Existing — Use the existing time balance

Note:

If you set a time balance other than None, you must set a skip property to tell Essbase what to do when it encounters missing values or values of 0. See “Selecting the Skip Option” on page 170.

6. Click Apply.

Note:

To restore the settings on this tab to their previously saved values, click Restore. To restore the settings to their original system default values, click Default.

**Time Balance Overview**

By default, Essbase bases the calculation of a parent of the time dimension on the consolidation properties and formulas of the children of the parent. For example, in the Year dimension, the Qtr1 member is, by default, the sum of the children January, February, and March. If, instead of a sum, you need a beginning, an ending, or an average value for a time period, you can select the appropriate time balance property for the appropriate member of the accounts dimension.
When you set a time balance property on a member of the accounts dimension, the property affects how the accounts member is calculated across the time dimension.

For example, if you want Opening Inventory for Qtr 1 to reflect the beginning inventory value for Qtr 1, set a time balance property of first on the accounts member Opening Inventory. When Opening Inventory is calculated across the Year dimension, Qtr 1 becomes equal to Jan (the first child in the Qtr 1 hierarchy).

**Selecting the Variance Reporting Method**

Variance reporting properties determine how Essbase calculates the difference between actual and budget data for a member whose formula includes an @VAR or @VARPER function. Any member that represents an expense to the company requires an expense property.

To select a variance reporting method:

2. Select the accounts dimension.
3. Select the Account Info tab.
4. Find the Variance Reporting grouping in the center-left of the dialog box.
5. Select a Variance Reporting method:
   - **Non Expense**—For non expense items, such as sales, actual should be greater than budget. When actual is less than budget, variance is negative. The @VAR function calculates ACTUAL - BUDGET. For example, if budgeted sales are $100 and actual sales are $110, the variance is 10. By default, members are non-expense.
   - **Expense**—For expense items, actual expenses should be less than budgeted expenses. When actual expenses are greater than budgeted expenses, variance is negative. The @VAR function calculates BUDGET - ACTUAL. For example, if budgeted expenses are $100 and actual expenses are $110, the variance is -10.
6. Click Apply.

**Note:**

To restore the settings on this tab to their previously saved values, click Restore. To restore the settings to their original system default values, click Default.

**Selecting the Skip Option**

If you set a time balance as first, last, or average, you must set a skip property to tell Essbase what to do when it encounters missing values or values of 0.

To select a skip option:

2 Select the accounts dimension.

3 Select the Account Info tab.

4 Find the Skip grouping in the top-right of the dialog box.

5 Select a Skip option:
   - None—Does not skip data when calculating the parent value. This property is the default value. If, however, Essbase encounters #MISSING data when calculating an average, it does not divide by the total number of members. It divides by the number of members with actual values. Therefore, setting the skip property to none or #MISSING does not affect average (but does affect first and last).
   - Missing—Skips #MISSING data when calculating the parent value.
   - Zero—Skips data that equals zero when calculating the parent value.
   - Missing and Zero—Skips #MISSING data and data that equals zero when calculating the parent value.

6 Click Apply.

Note:
To restore the settings on this tab to their previously saved values, click Restore. To restore the settings to their original system default values, click Default.

Adding Formulas to Dimensions

A formula determines how Essbase Server calculates the relationships between members of an Essbase database.

For more information on formulas, see “Formulas Overview” on page 172.

For information on dimensions, see “Dimensions Overview” on page 160.

➤ To add a formula to a dimension:

1 Access the dimension properties tabs of the Essbase Model Properties dialog box using the procedure in “Accessing the Dimension Properties Tabs” on page 161.

2 Select the Formula tab.

3 In the Formula text box, type a formula.

Note the following restrictions for formulas:
   - The length of the formula cannot exceed 64,000 characters.
   - Formulas in ASO (aggregate storage) models must be in MDX format.
   - Formulas in BSO (block storage) models must be in the standard Essbase calculation script format.
Formulas are not verified until the model is deployed.

4 **Click Apply.**

Note:

To restore the settings on this tab to their previously saved values, click Restore. To restore the settings to their original system default values, click Default.

**Formulas Overview**

A formula is a combination of operators and functions as well as dimension names, member names, and numeric constants. You can associate formulas with dimensions, user-defined members, and measures.

In calc formulas, member names used in the formula must be enclosed in double quotation marks (" "). End the formula with a semicolon (;). For example, enter "PROFIT" % "SALES";

When working with an aggregate storage outline, formulas must be expressed in MDX format. Use the appropriate MDX syntax; for example, (PROFIT*100)/SALES. The % operator is not supported in MDX formulas.

Note:

If the data storage model property for the outline is aggregate storage, you can specify only formulas for members within a dynamic hierarchy.

**Displaying Dimension Aliases**

In the Alias tab, you can see a list of current aliases for a dimension.

For information on aliases, see "Aliases Overview" on page 193.

For information on creating aliases, see “Creating Alias Sets” on page 114.

Note:

This is a read-only tab. To modify the alias assignments, see “Selecting Tables as Alias Sets” on page 153.

For information on dimensions, see “Dimensions Overview” on page 160.

To display the assigned aliases for a dimension:

1 Access the dimension properties tabs of the Essbase Model Properties dialog box using the procedure in “Accessing the Dimension Properties Tabs” on page 161.

2 Select the Alias tab.
A list of aliases assigned to the dimension is displayed.

**Assigning User-Defined Attributes to Dimensions**

You can assign a user-defined attribute (UDA) to a dimension to describe a characteristic of the dimension; for example, you might create a UDA called Big Market to identify any market that sells more than a certain amount:

See “UDAS Overview” on page 173.

See “Dimensions Overview” on page 160.

➤ To assign a UDA to a dimension:


2. Select the UDAs tab.

3. In the UDA value text box, enter a new UDA to use for the dimension.

   UDAs currently assigned to the dimension are displayed in the Existing UDAs text box.

4. Click Add to list.

   The UDA is displayed in the Existing UDAs box.

   **Note:**

   UDAs previously created are also displayed in the Existing UDAs box.

5. Click Apply.

   **Note:**

   To restore the settings on this tab to their previously saved values, click Restore. To restore the settings to their original system default values, click Default.

**UDAS Overview**

You can create your own UDAs (user-defined attributes) for dimensions and members. A UDA is a word or phrase; for example, you might create a UDA called Big Market to identify any market that sells more than a certain amount. Use UDAs in calculation scripts and report scripts.

When creating UDAs for members or dimensions, keep the following guidelines in mind:

- You can define multiple UDAs on a single member or dimension.
- You cannot define the same UDA twice on the same member or dimension.
- You can define the same UDA on different members and dimensions.
- A UDA can use the same name as a member, alias, level, or generation. When you name UDAs, follow the naming conventions in Appendix C, “Naming Restrictions for Essbase Studio”.
You cannot create a UDA on Essbase shared members.

A UDA applies to only a specified member. Descendants and ancestors of the member do not automatically receive the UDA.

Selecting Outline Build Options

Select the Outline Build tab to specify storage properties for alternate hierarchies when building Essbase outlines. You also use the Outline Build tab to move duplicate member settings in the Essbase outline.

For information on hierarchies, see “Hierarchies Overview” on page 119.

For information on dimensions, see “Dimensions Overview” on page 160.

The following topics describe the dimension properties you edit in the Outline Build tab:

- “Selecting Alternate Hierarchy Storage Settings” on page 174
- “Moving Duplicate Member Settings” on page 175
- “Optimizing Data Loads” on page 175

Selecting Alternate Hierarchy Storage Settings

You can specify storage properties for alternate hierarchies in the building of Essbase outlines.

To specify the storage option for alternate hierarchies:


2. Select the Outline Build tab.

3. Find the Hierarchy settings grouping near the top of the dialog box.

4. Select a hierarchy setting:
   - Stored at dimension level—Select to set the dimension as a stored hierarchy. This is the default.
   - Dynamic at dimension level—Select to set the dimension as a dynamic hierarchy.
   - Multiple-hierarchy enabled—Select to use both stored and dynamic hierarchies in the dimension.

5. Click Apply.

Note:

Because hierarchies determine how data is consolidated and navigated, you should review any ASO settings you may have for generation 2 members by clicking the link to the member info tab.
Moving Duplicate Member Settings
You can move duplicate member settings in the Essbase outline.

To move duplicate member settings:
1. Access the dimension properties tabs of the **Essbase Model Properties** dialog box using the procedure in “Accessing the Dimension Properties Tabs” on page 161.
2. Select the **Outline Build** tab.
3. Find the Move duplicate member settings grouping at the center of the dialog box.
4. Select the option for moving duplicate member settings:
   - **Do not move duplicate members**—Duplicate member settings will not be moved in the Essbase outline.
   - **Move duplicate members (and their descendants)**—Duplicate member settings will be moved in the Essbase outline.
5. Click **Apply**.

Note:
To restore the settings on this tab to their previously saved values, click Restore. To restore the settings to their original system default values, click Default.

Optimizing Data Loads
You can choose to optimize your data loads, a process which ignores any joins and, instead, uses the fact table columns as the data source bindings.

Note:
If you choose to optimize data loads, you should be aware of the possible effects on tables selected as alias tables. See “Selecting Tables as Alias Sets” on page 153.

To optimize data loads:
1. Access the dimension properties tabs of the **Essbase Model Properties** dialog box using the procedure in “Accessing the Dimension Properties Tabs” on page 161.
2. Select the **Outline Build** tab.
3. Find the **Optimize data load** grouping near the bottom of the dialog box.
4. Select **Optimize data load**.
Member Properties

The following topics discuss the member properties that you edit in each tab of the Essbase model properties dialog box at the member level:

- General tab—“Setting General Member Properties” on page 177
- Info tab—“Editing Member Information” on page 182
- Account Info tab—“Editing Members in Accounts Dimensions” on page 188
- Formula tab—“Adding Formulas to Members” on page 192
- Alias tab—“Editing Member Aliases” on page 192
- UDAs tab—“Assigning User-Defined Attributes to Members” on page 200
- Transformations tab—“Transforming Members” on page 200

For information on members, see “Members Overview” on page 176.

To access the member properties tabs, see “Accessing the Member Properties Tabs” on page 176.

Members Overview

Dimensions represent the core components of a business plan and often relate to business functions. Product, Region, and Year are typical dimensions. In most databases, dimensions are static, rarely changing over the life of the application.

A member is an individual component of a dimension. For example, Product A, Product B, and Product C might be members of the Product dimension. Each member has a unique name. A dimension can contain an unlimited number of members.

In some dimensions, members change frequently over the life of the application. Simultaneously, members can be parents of some members and children of other members. The Essbase outline indents members below one another to indicate a consolidation relationship. For example, sales totals for the Products dimension might be totaled by product description, broken down by product code, and further broken down by product ID.

To access the member properties tabs, see “Accessing the Member Properties Tabs” on page 176.

Accessing the Member Properties Tabs

To access the member properties tabs of the Essbase Model Properties dialog box:

1. In the Metadata Navigator, right-click the name of the model.
2 Select **Essbase Properties**.
3 Expand the model to display the names of the members.
4 Select a member.

The general tab for members is displayed by default.

**Setting General Member Properties**

In the General tab, you can set the general properties of members. You can also set attributes for the member, including varying attributes.

For information on members, see “Members Overview” on page 176.

The following topics describe the member properties you edit in the General tab:

- “Specifying General Member Properties” on page 177
- “Selecting Members as Attributes” on page 178
- “Setting Varying Attributes for Members” on page 178

**Specifying General Member Properties**

➤ To specify the general properties for a member in an Essbase model:

1 **Access the member properties tabs of the Essbase Model Properties dialog box using the procedure in “Accessing the Member Properties Tabs” on page 176.**

   The Name text box displays the name of the selected member.

   The Source element path text box displays the path of the currently selected member.

   The Data source binding text box displays the physical element with which the logical member is associated.

2 **Optional:** Add a comment or select a comment from the drop-down list.

   A comment for a Region member might be: Continental US, AK and HI. The drop-down list contains comments existing in the applicable database column.

   **Note:**

   For members of the Accounts dimension, the drop-down list box is not available when the Accounts dimension was created from the fact table.

3 **Optional:** In the **Member format string** text box, enter an MDX expression to reformat the values in cells so that the values are displayed in a desired format, for example, with a decimal point or a comma, or as text.

   **Note:**

   The member string format should not be confused with derived text measures. See “Creating or Editing Dimension Elements and Derived Text Measures” on page 100.
Note:
The MDX expression cannot be longer than 256 characters.

4 Click Apply.

Note:
To restore the settings on this tab to their previously saved values, click Restore. To restore the settings to their original system default values, click Default.

**Selecting Members as Attributes**

You can select a member as an attribute and specify the type of attribute.

For information on attributes and their use, see “Attributes Overview” on page 154.

➤ To select a member as an attribute:

1 Access the member properties tabs of the Essbase Model Properties dialog box using the procedure in “Accessing the Member Properties Tabs” on page 176.

2 Find the Attribute settings grouping in the center of the dialog box.

3 In Attribute settings, select the Essbase attribute for (member name) check box.

   The name of the dimension in which the member is located is displayed in the Attribute dimension name text box.

4 Select the attribute type from the drop-down list:
   ● Numeric
   ● Boolean
   ● String
   ● Date/Time

5 Click Apply.

Note:
To restore the settings on this tab to their previously saved values, click Restore. To restore the settings to their original system default values, click Default.

**Setting Varying Attributes for Members**

You can set varying attributes for members.

For information on varying attributes and their use, see “Varying Attributes Overview” on page 155.
Varying attributes are not supported in calendar hierarchies.

To set a member as a varying attribute:

1. Access the member properties tabs of the Essbase Model Properties dialog box using the procedure in “Accessing the Member Properties Tabs” on page 176.
2. Find the Attribute settings grouping in the center of the dialog box.
3. In Attribute settings, select the Essbase attribute for (member name) check box.
   - The name of the dimension in which the member is located is displayed in the Attribute dimension name text box.
   - By default, varying attributes for members are disabled.
4. From the drop-down list, select the Attribute type:
   - Numeric
   - Boolean
   - String
   - Date/Time
   - The default attribute type is String.
5. In the Varying Attribute Settings grouping, click the Edit button.
   - The Edit Varying Attributes dialog box is displayed.
6. Select the Create as Varying Attribute check box.
7. Select an Association mode:
   - Keep existing—If there are overlapping ranges for the attribute, the original range will be used to determine the value of the varying attribute.
   - Overwrite—If there are overlapping ranges for the attribute, the Overwrite option will cause the original range to be overwritten by the later range.
8. Find the Independent Dimension Settings grouping in the center of the dialog box.
9. In the Dimension column, select a dimension by selecting the check box next to the left of the dimension name.

A small white “X” in a red circle will appear in the State column, indicating that there are steps you must complete before the dimension is ready for varying attribute functions. If you hover your cursor over the “X,” a popup message will display the steps still required. If you perform a step that is not needed, the “X” will be replaced by a small, yellow triangle containing an exclamation point.
10 In the **Leaf Level** column, highlight the cell for the dimension and, from the drop-down list, select the member containing the range on which the value of the varying attribute will depend.

11 In the **Type** column, highlight the cell for the dimension and, from the drop-down list, choose whether the value of the varying attribute is **Range** or **Individual**:

- Range—the value of the varying attribute will reflect a range of values; for example, the sales representative for a client changes in midyear, so the value of the varying attribute covers January through December.

- Individual—the value of the varying attribute will reflect an individual value; for example, the sales representative for a client in Kansas is different from the sales representative for the client in Oklahoma, so the varying attribute covers Oklahoma only.

**Note:**

The type “Range” should be used for time ranges.

12 In the **From** column, click the cell for the dimension and click the button.

**Note:**

The expression to define an independent dimension binding is written in CPL (Common Platform Language). The expression is a sequence of operands and operators following the language-defined syntax. Each expression returns a value, the type of which defines the type of the expression. See Appendix D, “CPL Reference”.

The Independent Dimension Binding dialog box is displayed. Here, you will define the binding of the independent dimension and add filters to further refine the definition.

**Note:**

For ease of use, you can expand the Independent Dimension Binding dialog box.

13 In the **Independent Dimension Binding** dialog box, create an expression to define the binding for the dimension:

**Note:**

For limitations on creating expressions, see “Independent Dimension Bindings Limitations” on page 267

a. Select the **Source** tab in the lower-left of the dialog box.
b. Expand the dimensions to display the members.
c. Select a member.
d. Use the right-direction arrow to move the connection string for the member to the **Expression** text box.
e. Select the **Functions** tab.
f. Expand the SQL level to display the function types.
g. Expand the function types to display the functions.
h. Select a function.
i. Use the right-direction arrow to move the function string to the Expression text box.
j. Select the Operators tab.
k. Expand the operator types to display the operators.
l. Select an operator.
m. Use the right-direction arrow to move the operator to the Expression text box.

14 **Optional**—Create an expression to add filters to further refine the binding definition:
   a. Select the Source tab in the lower-left of the dialog box.
   b. Expand the dimensions to display the members.
   c. Select a member.
   d. Use the right-direction arrow to move the connection string for the member to the Filter text box.
   e. Select the Functions tab.
   f. Expand the SQL level to display the function types.
   g. Expand the function types to display the functions.
   h. Select a function.
   i. Use the right-direction arrow to move the function string to the Filter text box.
   j. Select the Operators tab.
   k. Expand the operator types to display the operators.
   l. Select an operator.
   m. Use the right-direction arrow to move the operator to the Filter text box.

15 Click **OK**.

16 **Optional**—In the **To** column, click the cell for the dimension and click the button which appears.

The Independent Dimension Binding dialog box is displayed. Here, you will continue to define the binding of the independent dimension and add filters to refine the binding.

**Note:**

For ease of you, you may want to expand the Independent Dimension Binding dialog box.

17 Repeat Steps 13 and 14 as needed.

18 In the **Edit Varying Attributes** dialog box, click **Ok**.
Editing Member Information

In the Info tab, you can edit a member so that data is calculated in the appropriate manner in your environment. You can specify how the data of children members is rolled up into their parents, and you can select the solve order of calculations. You can also select a two pass calculation to determine the value of children members whose values are dependent upon the values of their parents.

In the Info tab, you can select data storage methods to determine how and when Essbase stores data values for the member. Aggregate storage options can also be selected.

For information on members, see “Members Overview” on page 176.

The following topics describe the member properties you edit in the Info tab:

- “Selecting Consolidation Methods for Children Members” on page 182
- “Selecting the Member Calculation Solve Order” on page 183.
- “Selecting a Two Pass Calculation Option” on page 185
- “Selecting the Member Data Storage Method” on page 185
- “Selecting an Aggregate Storage Option” on page 186

Selecting Consolidation Methods for Children Members

Select a consolidation method to determine how children members will roll up into their parents during calculations.

For information on consolidation options, see “Consolidation of Children Members Overview” on page 183.

➤ To select a consolidation method:

1. Access the member properties tabs of the Essbase Model Properties dialog box using the procedure in “Accessing the Member Properties Tabs” on page 176.
2. Select the Info tab.
3. Find the Consolidation grouping at the top-left of the dialog box.
4. Select a Consolidation method:
   - + [addition]—Adds the member to the result of calculations performed on other members. This operator is the default operator if no other valid value is found.
   - - [subtraction]—Multiplies the member by -1 and then adds the result to the sum of calculations performed on other members.
   - * [multiplication]—Multiplies the member by the result of calculations performed on other members.
   - / [division]—Divides the member by the result of previous calculations performed on other members.
- **% [percent]**—Divides the member by the sum of previous calculations performed on other members and multiplies the result by 100 to yield a percentage value.
- **~ [ignore]**—Does not use the member in the consolidation to its parent.
- **∧ [never]**—Never uses the member for any consolidation.
- **External source**—In the drop-down list, select a column from the database where the consolidation operator is stored. The consolidation value (+, -, *, /, %, ~) of the selected column of the data source is used for consolidation.

5. **Click Apply.**

**Note:**
To restore the settings on this tab to their previously saved values, click Restore. To restore the settings to their original system default values, click Default.

### Consolidation of Children Members Overview

Member consolidation operators determine how children roll up into their parents. By default, new members are given the addition (+) operator, meaning that Essbase adds the members. For example, Essbase adds January, February, and March figures and stores the result in their parent, Qtr1.

**Important:** When working with an aggregate storage outline, the following consolidation property guidelines apply:
- For the dimension tagged as accounts, all consolidation operators can be used.
- For all other dimensions, only the (+) consolidation operator can be used.

Additionally, if the Label Only property is being used for members in a dimension, the direct child of the lowest Label Only member can have any consolidation operator; however, all indirect children of the lowest Label Only member must be tagged with the (+) consolidation property.

**Notes:**
- When working with aggregate storage outlines, use care when selecting a database column from the External source consolidation option. When performing loads, Essbase Studio Server may not recognize the consolidation operators that are stored in the database column. The server will not notify you of the potential for errors during validation.
- For members of the Accounts dimension, the From Database Column drop-down list box is not displayed when the Accounts dimension was created from the fact table.

### Selecting the Member Calculation Solve Order

The solve order determines the order by which members are evaluated in the dimension.

For information on using the solve order option, see “Solve Order Overview” on page 184.
To select a solve order:

1. Access the member properties tabs of the Essbase Model Properties dialog box using the procedure in “Accessing the Member Properties Tabs” on page 176.

2. Select the Info tab.

3. Find the Solve Order grouping in the left-center of the dialog box.

4. Select a Solve Order option:
   - Member Solve Order—In the text box, specify a number between 1 and 127 to represent the order in which the dynamic hierarchy is evaluated. The default is 0.
   - External Source—In the drop-down list, select the column in the database that contains the order in which members are evaluated.

   Note:
   The From Database Column drop-down list box is not available for members of user-defined dimensions or accounts dimensions that come from the fact table.

5. Click Apply.

   Note:
   To restore the settings on this tab to their previously saved values, click Restore. To restore the settings to their original system default values, click Default.

Solve Order Overview

The solve order number determines the order by which members are evaluated in the dimension. You can select a number between 1 and 127. The member with the highest solve order number is evaluated first (for example, a formula with a solve order number of 20 is evaluated before a formula with a solve order number of five).

Members with the same solve order number are evaluated in the order in which their dimensions appear in the database outline. Members with no solve order number are evaluated after all members with solve order members.

Note:
The default value is 0. Members with a solve order of 0 inherit the solve order of their dimension.

In the Dimension Solve Order text box, specify a number to represent the default order by which members are evaluated in the dimension. You can specify a solve order between 1 and 127. The default value is 0. In the Member Solve Order text box, specify the solve order for the dimension member. For example, in the Market dimension, Market is the dimension member.
Selecting a Two Pass Calculation Option

Two Pass Calculations are needed when the value of a child depends upon the value of the parent or the value of another member.

For information on using two pass calculation, see “Two Pass Calculation Overview” on page 166.

To select two pass calculation:

1. Access the member properties tabs of the Essbase Model Properties dialog box using the procedure in “Accessing the Member Properties Tabs” on page 176.
2. Select the Info tab.
3. Find the Two Pass calculation grouping at the top-right of the dialog box.
4. Select a Two Pass Calculation option:
   - None—Do not use two pass calculation on the currently selected member.
   - Two Pass Calculation—Use two pass calculation on the currently selected member.
   - External Source—In the drop-down list, select the column in the database that contains the two pass calculation.
5. Click Apply.

Note:
To restore the settings on this tab to their previously saved values, click Restore. To restore the settings to their original system default values, click Default.

Selecting the Member Data Storage Method

You can select a data storage method to determine how and when Essbase stores data values for a member. For example, you can tell Essbase to calculate the value for a member only when a user requests it and then to discard the value.

To select a member data storage method:

1. Access the member properties tabs of the Essbase Model Properties dialog box using the procedure in “Accessing the Member Properties Tabs” on page 176.
2. Select the Info tab.
3. Find the Data Storage method grouping in the right-center of the dialog box.
4. Select a Data Storage option:
   - Store Data—Stores the data value with the member. This is the default.
   - Dynamic Calc and Store—Calculates the data value when a user requests it and then stores the data value.
   - Dynamic Calc—Calculates the data value when a user requests it and then discards the data value.
● **Never Share**—Does not allow members to be shared.

● **Label Only**—Creates a member that is used for navigation. A label-only member contains no data value.

**Note:**

If you are using an aggregate storage outline, note the guidelines for selecting the Label Only option. See “Aggregate Storage with Label Only Option” on page 186.

● **Existing**—Use the existing storage method.

● **External source**—In the drop-down list, select the column in the database that contains the storage method.

5 **Click Apply.**

**Note:**

To restore the settings on this tab to their previously saved values, click Restore. To restore the settings to their original system default values, click Default.

### Aggregate Storage with Label Only Option

If you select the Label Only option with aggregate storage members, consider the following guidelines:

● For the dimension tagged as Accounts, the Label Only property for members can be used for all member levels, including the dimension level, with the exception of the level 0 member.

● For dimensions other than the accounts dimension, all levels above a member that is tagged as Label Only, including the dimension level, must also be tagged as Label Only. Additionally, all indirect children must be tagged with the (+) consolidation property, with the exception of the level 0 member.

**Note:**

A level 0 member cannot be tagged Label Only.

● If a dimension contains alternate hierarchies and the Label Only property is being used in any of the hierarchies, the same member level in all hierarchies within the dimension must be set to Label Only.

### Selecting an Aggregate Storage Option

If you are using aggregate storage, you can select an option for storing the data values. See “Aggregate Storage Guidelines” on page 187.
To select an aggregate storage option:

1. Access the member properties tabs of the **Essbase Model Properties** dialog box using the procedure in “Accessing the Member Properties Tabs” on page 176.

2. Select the **Info** tab.

3. Find the ASO Storage Options grouping at the lower-left of the dialog box.

4. Select the **ASO Storage Options**:
   - **Store**—Data values are stored with the member. This is the default setting for new members.
   - **Dynamic**—Data values associated with the member are not calculated until requested by a user. After the calculation is completed, the data values are not stored and are discarded.
   - **External source**—In the drop-down list, select the column in the database that contains the aggregate storage option.

5. Click **Apply**.

**Note:**

To restore the settings on this tab to their previously saved values, click **Restore**. To restore the settings to their original system default values, click **Default**.

**Aggregate Storage Guidelines**

When selecting an aggregate storage option, consider the following guidelines.

**Note:**

For information on aggregate storage and its use, see “Creating, Calculating, and Managing Aggregate Storage Databases” in the *Oracle Essbase Database Administrator’s Guide*.

**Aggregate Storage Guidelines for Hierarchy Types**

- Formulas can be specified only on members in a dynamic hierarchy or in a comparison dimension.

- If a dimension contains alternate hierarchies and the Label Only option is used in any hierarchy, the same member level in all hierarchies in the dimension must be set to Label Only.

- In a dynamic hierarchy, only stored members can have attribute associations.

- All members of stored hierarchies must be tagged with the + [addition] consolidation method. The only exception to this is a member tagged as Label Only member and any direct child of a Label Only member.

- A shared member may be used only once in a stored hierarchy.

- The top member in a dimension with multiple hierarchies must be tagged as Label.

- A compression dimension must contain a single dynamic hierarchy.

- A single dynamic hierarchy must have at least one level 0 member without a formula.
**Aggregate Storage Guidelines for Storage Types**

- The product of the number of stored dimension levels in an aggregate storage outline must be less than $2^{32}$.
- Currency settings cannot be specified.

**Aggregate Storage Guidelines for Dimension Types**

- Attribute dimensions cannot be associated with an accounts dimension.
- In the accounts dimension, the Label Only property for members can be used for all member levels with the exception of member level 0.
- In all dimensions other than the accounts dimension, all the levels above a member tagged as Label Only must also be tagged as Label Only. All children of a member tagged as Label Only must be tagged with the + [addition] consolidation method. The only exception to this is the direct child of the member.

**Note:**

A level 0 member cannot be tagged as Label Only.

- In the accounts dimension, all consolidation methods are allowed.
- In all dimensions other than the accounts dimension, only the + [addition] consolidation method can be used. All children of a member tagged as Label Only must be tagged with the + [addition] consolidation method. The only exception to this is the direct child of the member.
- Formulas can be specified only on members in a dynamic hierarchy or in a comparison dimension.
- Dynamic Time Series settings may be used only with a time dimension.
- A compression dimension must contain a single dynamic hierarchy.
- A compression dimension must have at least one level 0 member without a formula.

**Editing Members in Accounts Dimensions**

In the Account Info tab, you can edit the properties of accounts members to select optimal time balancing, instruct Essbase how to process missing values or zeroes, and select a method for reporting variances between actual and budget data.

For information on members, see “Members Overview” on page 176.

The following topics describe the member properties that you edit on the Account Info tab:

- “Selecting the Time Balance” on page 189
- “Selecting a Skip Option” on page 189
- “Selecting a Variance Reporting Method” on page 190
- “Specifying Data Load Scaling” on page 191
Selecting the Time Balance

In the accounts dimension, you can select a time balance method to determine the calculation method of parent members.

For information on how time balances function, see “Time Balance Overview” on page 169.

To select a time balance:

1. Access the member properties tabs of the Essbase Model Properties dialog box using the procedure in “Accessing the Member Properties Tabs” on page 176.
2. Select the Accounts dimension.
3. Select the member.
4. Select the Account Info tab.
5. Find the Time Balance grouping at the top-left of the dialog box.
6. Select a Time Balance option:
   - None—The default value. Essbase rolls up parents in the time dimension in the usual way—the value of a parent is based on the formulas and consolidation properties of the children of the parent.
   - First—The parent value represents the value of the first member in the branch (often at the beginning of a time period).
   - Last—The parent value represents the value of the last member in the branch (often at the end of a time period).
   - Average—The parent value represents the average of the children values.
   - External source—In the drop-down list, select the column in the database that contains the desired time balance option.

   **Note:**

   If you set a time balance as first, last, or average, you must set a skip property to tell Essbase what to do when it encounters missing values or values of 0. See “Selecting the Skip Option” on page 170.
7. Click Apply.

   **Note:**

   To restore the settings on this tab to their previously saved values, click Restore. To restore the settings to their original system default values, click Default.

Selecting a Skip Option

If you set a time balance as first, last, or average, you must set a skip property to tell Essbase what to do when it encounters missing values or values of 0.
To select a skip option:

1. Access the member properties tabs of the Essbase Model Properties dialog box using the procedure in “Accessing the Member Properties Tabs” on page 176.
2. Select the Accounts dimension.
3. Select the member.
4. Select the Account Info tab.
5. Find the Skip grouping at the top-right of the dialog box.
6. Select a Skip option:
   - None—The default value. Does not skip data when calculating the parent value. If, however, Essbase encounters #MISSING data when calculating an average, it does not divide by the total number of members; it divides by the number of members with actual values. Therefore, setting the skip property to none or #MISSING does not affect average (but does affect first and last).
   - Missing—Skips #MISSING data when calculating the parent value.
   - Zero—Skips data that equals zero when calculating the parent value.
   - Missing and Zero—Skips #MISSING data and data that equals zero when calculating the parent value.
   - External source—In the drop-down list, select the column in the database that contains the desired skip option.
7. Click Apply.

Note:
To restore the settings on this tab to their previously saved values, click Restore. To restore the settings to their original system default values, click Default.

Selecting a Variance Reporting Method

Variance reporting properties determine how Essbase calculates the difference between actual and budget data for a member whose formula includes an @VAR or @VARPER function. Any member that represents an expense to the company requires an expense property.

To select a variance reporting method:

1. Access the member properties tabs of the Essbase Model Properties dialog box using the procedure in “Accessing the Member Properties Tabs” on page 176.
2. Select the Accounts dimension.
3. Select the member.
4. Select the Account Info tab.
5. Find the Variance Reporting grouping at the top-left of the dialog box.
6. Select a Variance Reporting method:
● **Non Expense**—For non-expense items, such as sales, actual should be greater than budget. When actual is less than budget, variance is negative. The @VAR function calculates \( \text{ACTUAL} - \text{BUDGET} \). For example, if budgeted sales are $100 and actual sales are $110, the variance is 10. By default, members are non-expense.

● **Expense**—For expense items, actual expenses should be less than budgeted expenses. When actual expenses are greater than budgeted expenses, variance is negative. The @VAR function calculates \( \text{BUDGET} - \text{ACTUAL} \). For example, if budgeted expenses are $100 and actual expenses are $110, the variance is -10.

● **External source**—In the drop-down list, select the column in the database that contains the desired variance option.

7 Click **Apply**.

**Note:**

To restore the settings on this tab to their previously saved values, click **Restore**. To restore the settings to their original system default values, click **Default**.

---

**Specifying Data Load Scaling**

You can scale data as Essbase Studio loads it into an Essbase database. For example, you may want to scale the data if the values in the data source and the values in the database use different types of measurement, such as when the data source tracks sales in hundreds but the Essbase database tracks sales in single units. In this case, you would multiply incoming values by 100.

To select data load scaling for a member:

1 Access the member properties tabs of the **Essbase Model Properties** dialog box using the procedure in “Accessing the Member Properties Tabs” on page 176.

2 Select the **Accounts** dimension.

3 Select the member.

4 Select the **Account Info** tab.

5 Find the Data load scaling grouping near the bottom of the dialog box.

6 Select the **Enable scaling** check box.

7 In the **Scaling factor** text box, enter the scaling value to use; for example, enter 10 to multiply the incoming value by 10.

8 Click **Apply**.

**Note:**

To restore the settings on this tab to their previously saved values, click **Restore**. To restore the settings to their original system default values, click **Default**.
Adding Formulas to Members

A formula determines how Essbase calculates the relationships between members of an Essbase database. See “Formulas Overview” on page 172 and “Members Overview” on page 176.

➤ To add a formula to a dimension:

1. Access the member properties tabs of the Essbase Model Properties dialog box using the procedure in “Accessing the Member Properties Tabs” on page 176.
2. Select the Formula tab.
3. Select an option to enter a formula:
   - Enter a formula in the Formula text box.
   - Select From External source and from the drop-down list, select the external data source containing the formula you want to use.
4. Click Apply.

   Note:
   To restore the settings on this tab to their previously saved values, click Restore. To restore the settings to their original system default values, click Default.

Editing Member Aliases

In the Alias tab, you can see a list of alias sets created for a member. You can also change and reformat the names of the aliases.

For information and guidelines on using aliases, see “Aliases Overview” on page 193.

For information on members, see “Members Overview” on page 176.

The following topics describe the member properties you edit in the Alias tab:

- “Displaying Alias Sets” on page 194
- “Changing and Reformatting Alias Names” on page 194
- “Creating a Search Rule to Change and Reformat Alias Names” on page 194
- “Modifying a Search Rule to Change and Reformat Alias Names” on page 195
- “Changing Cases in Aliases” on page 196
- “Reformatting Spaces in Aliases” on page 197
- “Adding Prefixes to Aliases” on page 197
- “Adding Suffixes to Aliases” on page 199
Aliases Overview

An alias is an alternative, user-friendly name for a dimension or member. For example, a member identified by the SKU product code 100 could be given the more descriptive alias “Kool Cola.” This easily identifiable alias can then be displayed instead of the member name. The use of aliases thus improves the readability of Essbase outlines and reports.

The following topics are discussed in this overview:

- “Using Aliases” on page 193
- “Using Multiple Aliases” on page 193

Using Aliases

Key points when using aliases:

- Aliases are stored in alias tables.
- Aliases can be grouped by languages, regions, or descriptive names.
- For elements based in a relational source, you must create the alias column in the external data source in advance.
- Alias names are limited to 80 characters. If the column in the data source that contains the alias names has any value with more than 80 characters, you have two options:
  - Use a SUBSTRING function to extract the 80 or fewer characters you want to use in the alias name.
  - Replace the alias name.
- You can assign one or more aliases to a metadata element (in Essbase, a member level) by using alias sets. See “Working with Alias Sets” on page 113.

Using Multiple Aliases

Using alias sets, you can specify more than one alias at a member level. For example, you can use different aliases for different kinds of reports—users may be familiar with 100-10 as Cola, but advertisers and executives may be familiar with it as The Best Cola. This list illustrates how some products in a database can have two descriptive alias names:

<table>
<thead>
<tr>
<th>Product</th>
<th>Default</th>
<th>Long Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>100-10</td>
<td>Cola</td>
<td>The Best Cola</td>
</tr>
<tr>
<td>100-20</td>
<td>Diet Cola</td>
<td>Diet Cola with Honey</td>
</tr>
<tr>
<td>100-30</td>
<td>Caffeine Free</td>
<td>Cola All the Cola, none of the Caffeine</td>
</tr>
</tbody>
</table>

An alias set maps a specific, named set of alias names to member names. For each object in the logical tree, you assign one alias for each alias set defined in the catalog.

For information on alias sets, see “Alias Set Overview” on page 113.

You create your alias sets in the Alias Set Manager. See “Creating Alias Sets” on page 114.
Displaying Alias Sets

In the Alias tab, you can see a list of alias sets created for a member.

For information on aliases, see “Aliases Overview” on page 193.

For information on members, see “Members Overview” on page 176.

To display the assigned aliases for a member:

1. Access the member properties tabs of the Essbase Model Properties dialog box using the procedure in “Accessing the Member Properties Tabs” on page 176.
2. Select the Alias tab.
   
   A list of aliases assigned to the member will be displayed at the top of the dialog box.

Note:

The alias is read-only.

Changing and Reformatting Alias Names

You can rename and reformat alias names to improve the readability of outlines and reports. To do so, create or modify a search rule:

- See “Creating a Search Rule to Change and Reformat Alias Names” on page 194.
- See “Modifying a Search Rule to Change and Reformat Alias Names” on page 195.

Note:

When renaming aliases, you must follow the guidelines in Appendix C, “Naming Restrictions for Essbase Studio”.

For information on aliases, see “Aliases Overview” on page 193.

For information on members, see “Members Overview” on page 176.

Creating a Search Rule to Change and Reformat Alias Names

To create a search rule to change and reformat alias names:

1. Access the member properties tabs of the Essbase Model Properties dialog box using the procedure in “Accessing the Member Properties Tabs” on page 176.
2. Select the Alias tab.
3. Click the Create rule button.
   
   The Search Rule dialog box is displayed
4. In the Search for text box, enter the alias name you want to transform.
5 In the **Replace with** text box, enter the new alias name which will replace the current alias name.

6 Select the search options to refine the transformation:
   - **Case sensitive**—Only aliases that exactly match your search criteria will be transformed.
   - **Match whole word**—Only aliases that match your search criteria and are whole words will be transformed.
   - **Replace all occurrences**—All aliases that match your search criteria will be transformed.

   **Note:**
   You may select any, all, or none of the search options.

7 **Click OK to save your search rule.**

   The search rule that you created is displayed on the Alias tab. Note the icons that reflect the status of your search options:
   - ![Case sensitive](image)
     Case-sensitive search—“true” is displayed if you selected Case sensitive to refine your search.
   - ![Match whole word](image)
     Match whole words only—“true” is displayed if you selected Match whole word to refine your search.
   - ![Replace all occurrences](image)
     Replace all occurrences—“true” is displayed if you selected Replace all occurrences to refine your search.

8 **Click Apply.**

   **Note:**
   To restore the settings on this tab to their previously saved values, click Restore. To restore the settings to their original system default values, click Default.

---

**Modifying a Search Rule to Change and Reformat Alias Names**

To modify a search rule used to change and reformat alias names:

1 **Access the member properties tabs of the Essbase Model Properties dialog box using the procedure in “Accessing the Member Properties Tabs” on page 176.**

2 **Select the Alias tab.**

3 **In the Rules for text replacement section, right-click an alias.**

4 **Click the Edit details button.**

   The Search Rule dialog box is displayed.
5 Optional: In the Search for text box, enter an alias name that you want to transform, or use the alias name displayed in the text box.

6 In the Replace with text box, enter the new alias name which will replace the current alias name.

7 Select the search options to refine the transformation:
   - Case sensitive—Only aliases that exactly match your search criteria will be transformed.
   - Match whole word—Only aliases that match your search criteria and are whole words will be transformed.
   - Replace all occurrences—All aliases that match your search criteria will be transformed.

   Note:
   You may select any, all, or none of the search options.

8 Click OK to save your search rule.

   The search rule you created is displayed on the Alias tab. Note the icons which reflect the status of your search options:
   
   - Case-sensitive search—“true” is displayed if you selected Case sensitive to refine your search.
   - Match whole words only—“true” is displayed if you selected Match whole word to refine your search.
   - Replace all occurrences—“true” is displayed if you selected Replace all occurrences to refine your search.

9 Click Apply.

   Note:
   To restore the settings on this tab to their previously saved values, click Restore. To restore the settings to their original system default values, click Default.

Changing Cases in Aliases

You can change the case of an alias so that it is in all uppercase, all lowercase, or title case. You can also reset the case to its original format.

For information on aliases, see “Aliases Overview” on page 193.

For information on members, see “Members Overview” on page 176.
To change the case of an alias:

1. Access the member properties tabs of the *Essbase Model Properties* dialog box using the procedure in “Accessing the Member Properties Tabs” on page 176.

2. Select the *Alias* tab.

3. In the *Change Case to* drop-down list, select a case option.

4. Click *Apply*.

**Note:**

To restore the settings on this tab to their previously saved values, click *Restore*. To restore the settings to their original system default values, click *Default*.

### Reformatting Spaces in Aliases

You can reformat spaces in aliases by converting them to underscores. You can also drop leading and trailing spaces.

For information on aliases, see “Aliases Overview” on page 193.

For information on members, see “Members Overview” on page 176.

To reformat spaces in an alias name:

1. Access the member properties tabs of the *Essbase Model Properties* dialog box using the procedure in “Accessing the Member Properties Tabs” on page 176.

2. Select the *Alias* tab.

3. Select a spaces option:
   - Drop leading and trailing spaces—All leading and trailing spaces will be eliminated from the alias name.
   - Convert spaces to underscores—All spaces in the alias name will be changed to underscores.

4. Click *Apply*.

**Note:**

To restore the settings on this tab to their previously saved values, click *Restore*. To restore the settings to their original system default values, click *Default*.

### Adding Prefixes to Aliases

You can add a prefix or a suffix or both to an alias.

See “Adding Suffixes to Aliases” on page 199.

See “Aliases Overview” on page 193.

See “Members Overview” on page 176.
To add a prefix to an alias name:

1. Access the member properties tabs of the Essbase Model Properties dialog box using the procedure in “Accessing the Member Properties Tabs” on page 176.

2. Select the Alias tab.

3. Select a prefix option from the pull-down list:
   - None—No prefix will be added to the alias.
     
     Note:
     Select None if you plan to use a custom prefix as described in step 5, below.
   - Parent name—The name of the parent of the member will be added as a prefix to the alias name.
   - Grandparent name—The name of the grandparent of the member will be added as a prefix to the alias name.
   - All ancestors’ names—All names of all ancestors of the member will be added as a prefix to the alias name.
     
     Note:
     Selecting All ancestors’ names for a prefix may produce a lengthy prefix. Alias names are limited to 80 characters. See “Using Aliases” on page 193.
   - Dimension name—The name of the dimension in which the member is located will be added as a prefix to the alias name.

A preview of the alias name after the prefix is added is displayed in the text box just below the Custom prefix text box.

4. Optional: Select a Prefix separator to separate the prefix from the alias.

5. Optional: Enter a custom prefix in the Custom prefix text box.

   Note:
   If you enter a custom prefix, be sure you have selected None in the Prefix pull-down list as described in step 3, above.

   Note:
   Some special characters are not allowed in prefixes. See Appendix C, “Naming Restrictions for Essbase Studio”.

6. Click Apply.

   Note:
   To restore the settings on this tab to their previously saved values, click Restore. To restore the settings to their original system default values, click Default.
Adding Suffixes to Aliases

You can add a prefix, suffix, or both to an alias.

See “Adding Prefixes to Aliases” on page 197
See “Aliases Overview” on page 193.
See “Members Overview” on page 176.

To add a suffix to an alias name:

1. Access the member properties tabs of the Essbase Model Properties dialog box using the procedure in “Accessing the Member Properties Tabs” on page 176.
2. Select the Alias tab.
3. Select a suffix option from the pull-down list:
   - None—No suffix will be added to the alias
   - Parent name—The name of the parent of the member will be added as a suffix to the alias name.
   - Grandparent name—The name of the grandparent of the member will be added as a suffix to the alias name.
   - All ancestors’ names—All names of all ancestors of the member will be added as a suffix to the alias name.

   **Note:**
   Select None if you plan to use a custom suffix as described in step 5, below.

   - Parent name—The name of the parent of the member will be added as a suffix to the alias name.
   - Grandparent name—The name of the grandparent of the member will be added as a suffix to the alias name.
   - All ancestors’ names—All names of all ancestors of the member will be added as a suffix to the alias name.

   **Note:**
   Selecting All ancestors’ names for a suffix may produce a lengthy suffix. Alias names are limited to 80 characters. See “Using Aliases” on page 193.

   - Dimension name—The name of the dimension in which the member is located will be added as a suffix to the alias name.

   A preview of the alias name after the suffix is added is displayed in the text box just below the Custom suffix text box.

4. **Optional:** Select a Suffix separator to separate the suffix from the alias.
5. **Optional:** Enter a custom suffix in the Custom suffix text box.

   **Note:**
   If you enter a custom suffix, be sure you have selected None in the Suffix pull-down list as described in step 3, above.
Note:
Some special characters are not allowed in suffixes. See Appendix C, “Naming Restrictions for Essbase Studio”.

6 Click Apply.

Note:
To restore the settings on this tab to their previously saved values, click Restore. To restore the settings to their original system default values, click Default.

Assigning User-Defined Attributes to Members

You can assign a user-defined attribute (UDA) to a member to describe a characteristic of the member; for example, you might create a UDA called “Ounces.”

See “UDAs Overview” on page 173.
See “Members Overview” on page 176.

➤ To assign a UDA to a member:
1 Access the member properties tabs of the Essbase Model Properties dialog box using the procedure in “Accessing the Member Properties Tabs” on page 176.
2 Select the UDAs tab.
3 In the UDA value text box, enter a new UDA to use for the member.
4 Optional: Select External source and select the desired column in the drop-down list if you want Essbase Studio to retrieve the UDA from a column in the data source.
   UDAs assigned to the member are displayed in the Existing UDAs text box.
5 Click Apply.

Note:
To restore the settings on this tab to their previously saved values, click Restore. To restore the settings to their original system default values, click Default.

Transforming Members

In the Transformations tab, you can change and reformat the names of members to improve the readability of Essbase outlines and reports.

For information and guidelines on using aliases, see “Aliases Overview” on page 193.
For information on members, see “Members Overview” on page 176.
Because of the sequence which Essbase follows when transforming or reformatting member names, you may not be allowed to perform a transformation operation. See “Member Name Transformation Sequence” on page 201.
The following topics describe the member properties you edit in the Alias tab:

- “Adding Prefixes to Members” on page 201
- “Adding Suffixes to Members” on page 202
- “Changing and Reformatting Member Names” on page 204
- “Creating a Search Rule to Change and Reformat Member Names” on page 204
- “Modifying a Search Rule to Change and Reformat Member Names” on page 205
- “Changing Cases in Members” on page 206
- “Reformatting Spaces in Members” on page 207

**Member Name Transformation Sequence**

Essbase transforms and reformats member names in the following sequence:

1. Replacement strings are processed.
2. Leading and trailing spaces are dropped.
3. Spaces are converted to underscores.
4. Prefixes are added.
5. Suffixes are added.
6. Case changes are made.

**Note:**

Because Essbase follows this sequence, you are sometimes not allowed to perform a transformation. For example, if you add a custom prefix and select a replacement string for the prefix, the operation will not work because replacement strings are processed before prefixes are added.

**Adding Prefixes to Members**

You can add a prefix or a suffix or both to a member.

**Note:**

Prefixes and suffixes are not supported when the member is from a text file source.

To add a suffix to a member, see “Adding Suffixes to Members” on page 202.

For information on members, see “Members Overview” on page 176.

➤ To add a prefix to a member name:

1. **Access the member properties tabs of the Essbase Model Properties dialog box using the procedure in “Accessing the Member Properties Tabs” on page 176.**
2 Select the Transformations tab.

3 Select a prefix option from the pull-down list:
   - **None**—No prefix will be added to the member.
     
     **Note:**
     Select None if you plan to use a custom prefix as described in step 5, below.
   
   - **Parent name**—The name of the parent of the member will be added as a prefix to the member.
   
   - **Grandparent name**—The name of the grandparent of the member will be added as a prefix to the member.
   
   - **All ancestors’ names**—All names of all ancestors of the member will be added as a prefix to the member name.
     
     **Note:**
     Selecting All ancestors’ names for a prefix may produce a lengthy prefix. Member names are limited to 80 characters.
   
   - **Dimension name**—The name of the dimension in which the member is located will be added as a prefix to the member.

   A preview of the member name after the prefix is added is displayed in the text box just below the Custom prefix text box.

4 **Optional:** Select a Prefix separator to separate the prefix from the member.

5 **Optional:** Enter a custom prefix in the Custom prefix text box.

   **Note:**
   If you enter a custom prefix, be sure you have selected None in the Prefix pull-down list as described in step 3, above.

   **Note:**
   Some special characters are not allowed in prefixes. See Appendix C, “Naming Restrictions for Essbase Studio”.

6 **Click Apply.**

   **Note:**
   To restore the settings on this tab to their previously saved values, click Restore. To restore the settings to their original system default values, click Default.

**Adding Suffixes to Members**

You can add a prefix, suffix, or both to an member.
Note:
Prefixes and suffixes are not supported when the member is from a text file source.

To add a prefix to a member, see “Adding Prefixes to Members” on page 201.
For information on members, see “Members Overview” on page 176.

To add a suffix to a member name:

1. Access the member properties tabs of the Essbase Model Properties dialog box using the procedure in “Accessing the Member Properties Tabs” on page 176.
2. Select the Transformations tab.
3. Select a suffix option from the pull-down list:
   - None—No suffix will be added to the member.
   - Parent name—The name of the parent of the member will be added as a suffix to the member.
   - Grandparent name—The name of the grandparent of the member will be added as a suffix to the member.
   - All ancestors’ names—All names of all ancestors of the member will be added as a suffix to the member.

   Note:
   Select None if you plan to use a custom suffix as described in step 5, below.
   - Dimension name—The name of the dimension in which the member is located will be added as a suffix to the alias name.

   A preview of the member name after the suffix is added is displayed in the text box just below the Custom suffix text box.

4. Optional: Select a Suffix separator to separate the suffix from the alias.
5. Optional: Type a custom suffix in the Custom suffix text box.

   Note:
   If you enter a custom suffix, be sure you have selected None in the Suffix pull-down list as described in step 3, above.
Note:
Some special characters are not allowed in suffixes. See Appendix C, “Naming Restrictions for Essbase Studio”.

6 Click Apply.

Note:
To restore the settings on this tab to their previously saved values, click Restore. To restore the settings to their original system default values, click Default.

Changing and Reformatting Member Names
You can change or reformat member names to improve the readability of outlines and reports. To do so, create or modify a search rule:

● See “Creating a Search Rule to Change and Reformat Member Names” on page 204.
● See “Modifying a Search Rule to Change and Reformat Member Names” on page 205.

Note:
When renaming members, you must follow the guidelines in Appendix C, “Naming Restrictions for Essbase Studio”.

For information on members, see “Members Overview” on page 176.

Because of the sequence which Essbase follows when transforming or reformatting member names, you may not be allowed to perform a transformation operation. See “Member Name Transformation Sequence” on page 201.

Creating a Search Rule to Change and Reformat Member Names

To create a search rule to change and reformat member names:

1 Access the member properties tabs of the Essbase Model Properties dialog box using the procedure in “Accessing the Member Properties Tabs” on page 176.

2 Select the Transformations tab.

3 Click the Create rule button.

The Search Rule dialog box is displayed.

4 In the Search for text box, enter the member name that you want to transform.

5 In the Replace with text box, enter the new member name which will replace the current member name.

6 Select search options to refine the transformation:

● Case sensitive—Only members that exactly match your search criteria will be transformed.
● Match whole word—Only members that match your search criteria and are whole words will be transformed.
● Replace all occurrences—All members that match your search criteria will be transformed.

Note:
You may select any, all, or none of the search options.

7 Click OK to save your search rule.

The search rule you created is displayed on the Transformations tab. Note the icons which reflect the status of your search options:

● Case sensitive search—“true” is displayed if you selected Case sensitive to refine your search.
● Match whole words only—“true” is displayed if you selected Match whole word to refine your search.
● Replace all occurrences—“true” is displayed if you selected Replace all occurrences to refine your search.

8 Click Apply.

Note:
To restore the settings on this tab to their previously saved values, click Restore. To restore the settings to their original system default values, click Default.

Modifying a Search Rule to Change and Reformat Member Names

➢ To modify a search rule used to change and reformat member names:

1 Access the member properties tabs of the Essbase Model Properties dialog box using the procedure in “Accessing the Member Properties Tabs” on page 176.

2 Select the Transformations tab.

3 In the Rules for text replacement section, right-click a member.

4 Click the Edit details button.
The Search Rule dialog box is displayed.

5 Optional: In the Search for text box, enter a member name that you want to transform, or use the member name displayed in the text box.

6 In the Replace with text box, enter the new member name which will replace the current member name.

7 Select the search options to refine the transformation:
- **Case sensitive**—Only members that exactly match your search criteria will be transformed.
- **Match whole word**—Only members that match your search criteria and are whole words will be transformed.
- **Replace all occurrences**—All members that match your search criteria will be transformed.

**Note:**
You may select any, all, or none of the search options.

8 **Click OK to save your search rule.**

The search rule you created is displayed on the Transformations tab. Note the icons which reflect the status of your search options:

- **Case sensitive search**—“true” is displayed if you selected Case sensitive to refine your search.
- **Match whole words only**—“true” is displayed if you selected Match whole word to refine your search.
- **Replace all occurrences**—“true” is displayed if you selected Replace all occurrences to refine your search.

9 **Click Apply.**

**Note:**
To restore the settings on this tab to their previously saved values, click Restore. To restore the settings to their original system default values, click Default.

**Changing Cases in Members**

You can change the case of a member so that it is in all uppercase, all lowercase, or title case. You can also reset the case to its original format.

For information on members, see “Members Overview” on page 176.

➤ **To change the case of a member:**

1. Access the member properties tabs of the **Essbase Model Properties** dialog box using the procedure in “Accessing the Member Properties Tabs” on page 176.
2. Select the **Transformations** tab.
3. In the **Change Case to** drop-down list, select a case option.
4. Click Apply.
Note:
To restore the settings on this tab to their previously saved values, click Restore. To restore the settings to their original system default values, click Default.

Reformatting Spaces in Members
You can reformat spaces in members by converting them to underscores. You can also drop leading and trailing spaces.

For information on members, see “Members Overview” on page 176.

➢ To reformat spaces in a member name:

1 Access the member properties tabs of the Essbase Model Properties dialog box using the procedure in “Accessing the Member Properties Tabs” on page 176.
2 Select the Transformations tab.
3 Select a spaces option:
   ● Drop leading and trailing spaces—All leading and trailing spaces will be eliminated from the member name.
   ● Convert spaces to underscores—All spaces in the member name will be changed to underscores.
4 Click Apply.

Note:
To restore the settings on this tab to their previously saved values, click Restore. To restore the settings to their original system default values, click Default.

Working with Essbase Model Properties
After you have edited the Essbase model properties, you can perform the following tasks:

● “Viewing Models” on page 207
● “Validating Model Properties” on page 209
● “Reviewing Changes to Properties” on page 210
● “Browsing Models” on page 210
● “Using Tool Tips” on page 211
See “Models Overview” on page 146.

Viewing Models
You can use the model work area to see a graphical representation of a model. This function will help you see the hierarchical structure of the model.
You can also use the model work area to launch the Cube Deployment Wizard and to edit model properties.

For information on models, see “Models Overview” on page 146.

The following topics discuss the tasks you can perform in the model work area:

- “Opening the Model Work Area” on page 208
- “Using the Model Work Area” on page 208

**Opening the Model Work Area**

You can use the model work area to see a graphical representation of dimensions and members.

To open the model work area:

1. In the Metadata Navigator, right-click the name of the model.
2. Select Edit.

A graphical illustration of the model with dimensions and members is displayed in the model work area as seen below:

![Model Work Area](image)

**Using the Model Work Area**

The model work area allows you to perform the following tasks:

- **Move dimensions**: To move a dimension in the model work area, left-click the dimension and drag it to the desired location.
- **Expand and collapse dimensions**: To expand and collapse a dimension to see or hide members, left-click on the maximize and minimize icons in the top-right corner of the dimension.
- **Undo**: To undo a dimension move, right-click in the model work area and select Undo.
- **Redo**: To redo a dimension move, right-click in the model work area and select Redo.
● **Zoom In:** To get a larger view of a dimension, right-click in the model work area and select Zoom In.

● **Zoom Out:** To get a smaller view of a dimension, right-click in the model work area and select Zoom Out.

● **Launch the Cube Deployment Wizard:** Right-click in the model work area and select Cube Deployment Wizard.

  See “Deploying Cubes” on page 214.

● **Edit properties:** To edit model, dimension, and member properties, right-click in the model work area and select Essbase Properties.

### Validating Model Properties

You can use the Validate Properties dialog box to check for errors in your model such as a missing required dimension or invalid name.

For information on models, see “Models Overview” on page 146.

➤ To validate a model:

1. Access the **Essbase Model Properties** dialog box using the procedure in “Accessing the Model Properties Dialog Box” on page 147.

2. Click the **Run validation** button, ![run validation button](image), to access the **Validate Properties** dialog box.

3. Double-click one of the displayed elements (model, dimension, or member) to see the errors or invalid properties associated with the element.

**Note:**

If no elements are displayed in the Validate Properties dialog box, no error conditions exist in your model.

4. Select the error message and right-click to display a menu of options, offering possible fixes to the error.

5. Select an option.

  The properties dialog box for correcting the error is displayed.

6. Correct the error in the properties dialog box, and click **Apply**.

7. Click the **Revalidate** button, ![revalidate button](image), to see if the error has been corrected.

**Note:**

If the error has been corrected, the element will no longer be displayed in the dialog box.

8. When all errors have been corrected, click **Close**.
Reviewing Changes to Properties

You can use the Applied Properties dialog box to see a list of changes that you have applied to the model properties in this session.

Note:

Only the changes you have made this session are shown. A session is defined as the time during which you have accessed the Essbase Properties tabs, lasting until you close the tabs.

For information on models, see “Models Overview” on page 146.

To see a list of changes to the model properties:

1. Access the Essbase Model Properties dialog box using the procedure in “Accessing the Model Properties Dialog Box” on page 147.

2. Click the Display changes button, , to access the Applied Properties dialog box.
   The changes made to your model during this session are displayed.

3. Expand one of the displayed elements (model, dimension, or member) to see the change made to the element.

4. If you want to modify the change, expand the element.
   The properties dialog box for changing the property is displayed.

5. In the properties dialog box, edit the property and click Apply.

6. Click the Display changes button, , to access the Applied Properties dialog box.
   The new changes made to your model during this session are displayed.

7. When you have finished reviewing your changes, click Cancel.

Browsing Models

You can use the Back To and Forward To buttons to return to a dimension or member that you viewed or edited previously.

Note:

Only the dimensions and members that you have viewed or edited this session are shown. A session is defined as the time during which you have accessed the Essbase Properties tabs, lasting until you close the tabs.

For information on models, see “Models Overview” on page 146.
To browse dimensions and members that you have viewed or edited:

1. Access the Essbase Model Properties dialog box using the procedure in “Accessing the Model Properties Dialog Box” on page 147.
2. Expand the model to display the names of the dimensions and members.
3. Select a dimension or member name.
   The properties tabs for the dimension or member are displayed.
4. Optional: Edit the dimension or member properties and click Apply.
5. If after editing or viewing the properties of several dimensions and members, to return to a previous dimension or member that you viewed or edited, click the Back To button, 
   The immediately previous dimension or member is displayed with the applicable properties tabs.
6. To return to the next previous dimension or member that you viewed or edited, click the Back To button, again.
   The dimension or member is displayed with the applicable properties tabs.
7. To go forward through the dimensions and members that you have viewed or edited, click the Forward To button, 
   The next dimension or member is displayed with the applicable properties tabs.

Using Tool Tips

You can use your cursor to view tool tips, providing a quick view of model, dimension, and member properties.

To get a quick view of properties using tool tips:

1. Access the Essbase Model Properties dialog box using the procedure in “Accessing the Model Properties Dialog Box” on page 147.
2. Expand the model to display the names of the dimensions and members.
3. Place your cursor over the name of the model or a dimension or member.
   A summary of key properties of the dimension or member is displayed.
Cube Deployment Overview

The Essbase models you build with the Cube Schema Wizard are used to deploy cubes to Essbase Server. Cube deployment is a process of setting load options in order to build an outline and load data into an application and database. The outline you build is editable in Administration Services Console. The application and database you create is accessible from Smart View and Essbase Spreadsheet Add-in.

Using the Cube Deployment Wizard, you can choose from several load options such as when you deploy a cube:

- Build outline only
- Load data only
- Build outline and load data
- Load members incrementally
- Delete members from an existing cube before loading
- Delete and restore data to an existing cube

You can also choose to create and save a rules file only during deployment, then access Administration Services Console to load members or data or both at a later time.

If you did not choose to create an Essbase model option during the cube schema creation process, you can create one at any time. You can also create more than one Essbase model for the same cube schema. See “Creating Essbase Models from Existing Cube Schemas” on page 142) for instruction. After creating the Essbase model, launch the Cube Deployment Wizard and complete the first page, as described in “Providing Connection and Load Information for Cube Deployment” on page 215. From the wizard, you can also modify Essbase properties for the model, as described in Chapter 11, “Essbase Properties.”
Note:
This chapter describes deploying cubes using the Cube Deployment Wizard. You may also deploy cubes using the MaxL deploy command. All the options that are available in the Cube Deployment Wizard are also available in the deploy command. See the MaxL documentation in the Oracle Essbase Technical Reference for information.

Deploying Cubes

The Cube Deployment Wizard consists of two screens:

- **Essbase Server Information**—you provide information regarding the Essbase Server connection, the type of load you are performing, and any load options. See “Providing Connection and Load Information for Cube Deployment” on page 215.

  Note:
The first time you deploy a cube to an instance of Essbase Server, you are prompted for Essbase Server connection information, as described in “Creating a Connection to Essbase Server” on page 214. After you provide this information, the Cube Deployment Wizard is launched.

- **Incremental Load**—If you choose to load members incrementally, you can select which members to update. See “Setting Up an Incremental Load for Cube Deployment” on page 218.

You can also review the history of cube deployments for each Essbase model, as described in “Viewing Deployment History” on page 219.

Creating a Connection to Essbase Server

The first time you deploy a cube to an instance of Essbase Server, you are prompted for Essbase connection information.

➤ To create a connection to Essbase Server:

1. In the Metadata Navigator, select the Essbase model from which you want to deploy a cube.

2. Right-click the model name and select **Cube Deployment Wizard** from the popup menu.

   The Essbase Login dialog box is displayed, where you will create a reusable connection to Essbase.

   Note:
   If there is already an existing Essbase connection, the Cube Deployment Wizard is displayed, not the Essbase Login dialog box.

3. In **Essbase Login**, provide the following information, and then click **Login**:

   - **Name** for this connection
• An optional Description.

• The Server name and the Port number to which you want to deploy; for example, aspen5.
  
  The default for Server and Port is the instance of Essbase running on the same machine as Essbase Studio.

• The name of the User and the Password needed to access this Essbase Server.

Note:

The connection name you provide is selectable now and during future deployments in the Essbase Server Connection field of the Cube Deployment Wizard.

Providing Connection and Load Information for Cube Deployment

Use this page of the Cube Deployment Wizard to specify connection information and load parameters. You can also modify Essbase model properties before deployment.

➤ To set up server, load, and model properties information for cube deployment:

1 In the Metadata Navigator, select the Essbase model from which you want to deploy a cube.

2 Right-click the model name and select Cube Deployment Wizard from the popup menu.

3 In Essbase Server Connection, select the connection representing the Essbase Server to which you want to deploy.

If you want to deploy to an Essbase Server connection that is not on the list, click New Connection and complete the steps in “Creating a Connection to Essbase Server” on page 214, then continue with step 4.

4 Enter the Application and Database name to which you are deploying.

Note:

If you are deploying from an Essbase model that is enabled for XOLAP, you must specify a new application and database name each time you deploy. You cannot deploy over an existing application and database.

5 In Rejected records settings, perform these tasks:

a. For Number of records to keep, choose an option:
  
   • All—To keep all rejected records in the error file
   
   • Limit—To keep the number of rejected records in the error file that you specify in the text box at the right. The default is keep 200 records.

b. For Error file name, choose an option:
  
   • Default—To give the error file the default file name. The default file name has the following format:

   app_name.db_name_timestamp.err
For example:
myesbapp.myesbdb_05-02-08_11_30-38.err

- **File name**—To specify a custom file name. For example, you may want to name the error file, `errors.err`.

You may also specify a different location for the error file. By default, the error file is placed in:

`HYPERION_HOME/Essbase/EssbaseStudio/server/essjapihome/data`

To place the error file in a location other than the default, enter the relative path in the “File name” text box. For example, to name the error file `errortest`, and place the file in a folder called `errors` under the `essjapihome/data` directory, enter the following:

`errors/errortest.err`

To name the error file `testerrors` and place the file in a folder called `errorlogs` under the Essbase Studio `server` directory, enter the following:

`../../../errorlogs/testerrors.err`

6. **Optional:** Select **Abort data load when the first error is rejected** to stop a data load after Essbase Studio encounters the first error during the load process.

This option is enabled when the “Load data” or “Build outline and load data” option is selected.

7. In the **Load task type** group, choose one of the following load options:

- **Build outline**—adds new dimensions or members (without data) to an Essbase outline
- **Load data**—populates an Essbase database with data. Loading data establishes actual values for the cells defined by the structural outline of the database.
- **Build outline and load data**—adds new dimensions or members without data to an Essbase outline and populates an Essbase database with data.

8. In the **Load data options** group, select one of the following options:

- **Add to existing data**—Select to add values in the data source to the existing values in the cube.
- **Subtract from existing data**—Select to subtract the values in the data source from the existing values in the cube.
- **Overwrite existing data**—Select to replace the values in the cube with the values in the data source.

9. **Optional:** Select **Delete all members first** if you want to delete all dimensions and members in an existing Essbase outline.

When you delete all members, Essbase Studio removes all members from the existing Essbase database outline and then uses the member levels of the cube schema to recreate the outline. Because deleting all members can be slower than creating or updating an Essbase outline without deleting all members, Oracle recommends using this option only if you have a specific reason to do so. You should delete all members if, for example, you know that some members have
been removed from an existing Essbase model and you want to build a smaller Essbase outline that contains the smaller set of members.

This check box is not enabled when you choose to only load data.

10 **Optional:** Select **Delete and restore database** to delete all members and data in the Essbase database before performing a member load or a member and data load.

This action clears the Essbase database outline of members and data before the outline build occurs, significantly reducing the amount of time required for the load.

This check box is not enabled when you choose to only load data.

11 **Optional:** Select **Incremental Load** to select specific dimensions or members to update in the Essbase outline.

This check box is not enabled when you choose to only load data.

**Note:**
When you select this option, click the Next button and complete the tasks in “Setting Up an Incremental Load for Cube Deployment” on page 218.

12 **Optional:** Select **Create and save rule file only** to specify the changes Essbase should make to data and members from a data source while loading them into the Essbase database.

The data source is not changed.

**About Rules and Rules Files**

Rules define operations that Essbase performs on data values or on dimensions and members when it processes a data source. Use rules to map data values to an Essbase database or to map dimensions and members to an Essbase outline.

Rules are stored in rules files. A rules file defines which build method to use, whether data values or members are sorted or in random order, and how to transform data values or members before loading them. Essbase Studio creates a separate rules file for each dimension.

Rules files are saved to the app directory of your Essbase installation.

13 **Optional:** Click **Model Properties** if you want to modify the Essbase properties of the model.

For information on modifying Essbase model properties, see Chapter 11, “Essbase Properties.”

14 **Click OK to deploy the cube.**

Deployed cubes are displayed in the Metadata Navigator as children of the Essbase model from which they were deployed.

After deployment, launch Administration Services Console to view and edit the outline. You can also query the cube using Smart View or Essbase Spreadsheet Add-in.

**Note:**
If you selected the Incremental Load option, click Next and complete the tasks in “Setting Up an Incremental Load for Cube Deployment” on page 218.
Setting Up an Incremental Load for Cube Deployment

In the Incremental Load page of the Cube Deployment Wizard, you select specific dimensions or members to update.

➤ To set member load update options:

1 In the Incremental Load page of the Cube Deployment Wizard, under Dimension Build Update, choose one of the following options:

● Update all hierarchies—All hierarchies in the model are updated; any new members are added.

When this option is selected, all hierarchies are automatically selected for update; and the Update, Rebuild, and Clear buttons are not available.

● Update or rebuild selected—When this option is selected, all hierarchy selections are cleared. You then select a hierarchy, and then choose one of the following operations to perform on the hierarchy:

  ○ Update—Add any new members to the hierarchy; do not delete existing members.

  This option should be used to add new members without changing the hierarchy's structure, or to add shared members. During Update, an existing hierarchy is updated without removing the existing members.

  When you select Update, an icon is displayed next to each member in the hierarchy signifying that these members are marked for Update. The check boxes next to each member are not yet selected. You must manually select them.

  ○ Rebuild—Clear all the members of the hierarchy and add back all members, including shared members. If necessary, restructure the hierarchy.

  This option is particularly useful if you have removed members from a hierarchy. Then the members that still exist, plus any new ones, are added back into the hierarchy and, if necessary, the hierarchy is restructured.

  When you select Rebuild, an icon is displayed next to each member in the hierarchy signifying that these members are marked for Rebuild. The check boxes next to each member are not yet selected. You must manually select them.

  ○ Clear—Use the Clear button to clear any selections you have made during the current session. If you had marked a hierarchy in error, and then selected the check boxes for that hierarchy, click Clear to clear these selections.

Note:
The Clear button only works on hierarchies where the check box next to each of its members is selected.

2 To set the parameters for restructuring the database during member build, select one of the following Preserve options.

● All Existing Data—Preserves all existing data that applies to the changed outline when restructuring occurs. This is the default.
● **Input data**—Preserves 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 node) blocks.

● **Level 0 Data**—Preserves 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 recreated when you calculate the database.

● **No Data**—Clears all data from the database.

3 Click Finish to deploy the cube.

Deployed cubes are displayed in the Metadata Navigator as children of the Essbase model from which they were deployed.

You can now launch Administration Services Console to view and edit the outline. You can also query the cube using Smart View or Essbase Spreadsheet Add-in.

### Viewing Deployment History

You can view the deployment history of a given Essbase model to see a listing of all children cubes that were deployed to various instances of Essbase Server.

➢ To view the deployment history of cubes related to an Essbase model:

1 In the Metadata Navigator, navigate to the Essbase model whose history you want to view.

2 Right-click the model name and select Show Deployment History.

A tab called “Deployment History: <Essbase Model Name>” is displayed in the work area of Essbase Studio Console, listing each deployment in a grid format.

The information shown is:

- Status—Whether the deployment succeeded or failed
- Time—The date and time of the deployment
- Essbase Server—The Essbase Server instance name to which the cube was deployed
- Application—The Essbase application name of the cube
- Database—The Essbase database name of the cube
- Type—Whether the deployment was a member load, data load, or both
- User—The user name of the user who deployed the cube

3 Double-click a line in the grid representing a deployment to view the Deployment Errors and Warnings dialog box.

Any errors or warnings logged during deployment are listed here.
4 Click **OK** or **Cancel** to close **Deployment Errors and Warnings**.

5 When finished viewing, click the **X** in the tab of the **Deployment History** window to close it.
In This Chapter

Launching Administration Services Console ................................................................. 221
Updating Cube Linkage .............................................................................................. 222

Launching Administration Services Console

After you deploy an Essbase application, you can go directly from Essbase Studio to the Administration Services Console, where you can continue working with the application outline.

Once in Administration Services Console, access the Oracle Essbase Administration Services Online Help from the Help menu or from the Help buttons within the console dialog boxes.

Prerequisite to launching Administration Services Console from Essbase Studio Console for the first time, ensure that the parameters in the EASLaunch.properties file are properly configured. At a minimum, the EASHome and EASServer parameters must be set. The EASLaunch.properties file is located in the Essbase Studio installation in the console \eas directory. See “Configuring the Administration Services Console Launch File” on page 37 for instructions on setting the parameters in this file.

Note:

Before launching Administration Services Console, ensure that the instances of Essbase Server and Essbase Administration Server that you are planning to access are running.

➢ To launch Administration Services Console from Essbase Studio:

1 **In the Metadata Navigator**, locate the deployed application that you will be working with in Administration Services Console.
 Deployed applications are displayed under Essbase model names.

2 **Right-click the deployed application and select Launch Essbase Administration Services Console.**
 The Administration Services Console is opened.
 Depending on how you configured the EASLaunch.properties file, you may or may not be prompted for a user name and password. See “Configuring the Administration Services Console Launch File” on page 37 for more information.
3 If the Essbase Administration Services Login dialog box is displayed, enter your login name and password, and then click OK.

From this step forward, you are in the Administration Services Console user interface using Oracle Essbase Administration Services functionality. Access the Essbase Administration Services Online Help from the Help menu or by clicking the Help button in dialog boxes.

4 In the Enterprise View of Administration Services Console, locate the application with which you want to work.

You can now work on the outline in Administration Services Console.

### Updating Cube Linkage

Essbase cubes store information on the Essbase Studio Server and port used to deploy the cube, the model the cube was based on, and the login information for the data source from which the cube was built.

For an existing cube built using Essbase Studio, you can update cube linkage to change the Essbase Studio Server, port, and model context.

For example, drill-through reports can be redirected to a new Essbase Studio Server and can be based on a different Essbase model. This can be a useful load balancing tool.

You can also update cube linkage when the data source login information changes, which is especially helpful for XOLAP-enabled cubes. By updating the cube linkage, the data source login information is updated without recreating the model or redeploying the cube.

**Before you begin:**

- Be sure you have the correct Essbase Studio Server machine name and port number information.
- Ensure that the associated catalog for any Essbase Studio Server instances you enter contains the Essbase model names you want to link to.
- If the login information for the data source used to create the cube has changed, update the login information in the Properties dialog box for the appropriate data source connection, as described in “Editing Data Source Connection Properties” on page 71.

To update the cube linkage for a given cube built using Essbase Studio:

1 In Essbase Studio Console, select Tools, then Update Cube Linkage.

2 In Update Cube Linkage, under Essbase Connections, expand the applicable Essbase application and select the Essbase cube whose linkages you want to change.

   Note the entries under “Cube Linkage Essbase Studio Server” and “Cube Linkage Essbase Model” for the selected cube.

3 Click Update to launch the Cube Linkage dialog box.

4 Enter the new Essbase Studio Server name and port number in the following format:

   `essbase_studio_server_name:port_number`
For example:

aspen3:9080

**Note:**

Care must be taken when entering a new server name and port number. Essbase Studio does not validate this information.

Essbase Studio remembers the server name and port number combinations you have entered previously. If it is available in the drop-down list, you may select it.

**Note:**

If you are updating cube linkage only because login information for the data source changed, you do not need to enter a new Essbase Studio Server name or port. Enter a new server name or port only if you want to point to a new server or port.

5 Enter the **Essbase model name**.

Be sure that the Essbase model name you enter exists in the catalog that is associated with the Essbase Studio Server you entered in step 4.

**Note:**

Care must be taken when entering a new model name. Essbase Studio does not validate this information.

**Note:**

If you are updating cube linkage only because login information for the data source changed, you do not need to enter a new model name. Enter a new model name only if you want to point to a new model.

6 Click **OK** to return to **Update Cube Linkage**.

7 For the selected cube, verify that the entries under **Cube Linkage Essbase Studio Server** and **Cube Linkage Essbase Model** are correct.

8 Repeat step 2 through step 7 for all cubes to update linkage for

9 To exit, click **Close**.
In This Chapter

Lineage Overview.........................................................................................................................225
Opening the Lineage Work Area ..................................................................................................226
Using the Lineage Work Area ......................................................................................................226

Lineage Overview

The lineage work area is a graphical representation of the lineage of a metadata element, enabling you to see the relationships among metadata elements in the catalog.

The lineage work area enables you to view information ranging from which data source table and column the element comes from to the deployed cubes that the element is used in. The lineage work area can help you to understand more quickly the potential impact of changes to the element; for example, the effect of removing a hierarchy from the cube schema. You can also see the history of each element.

The lineage work area displays the chain of elements related to the selected element you have chosen. For example, a standard hierarchy would show these related objects:

- The source tables, including the key and caption for each table on which the hierarchy was built
- The columns that make up the hierarchy
- The members within the hierarchy, including user-defined members
- The cube schemas in which the hierarchy is used
- The Essbase models created from the cube schemas that use the hierarchy
- The Essbase cubes that were deployed from the Essbase models (created from the cube schemas that use the hierarchy)

To display the lineage of an element, see “Opening the Lineage Work Area” on page 226.

For information on optional tasks you can perform in the lineage work area, see “Using the Lineage Work Area” on page 226.
Opening the Lineage Work Area

The lineage work area is a graphical representation of the lineage of a metadata element, enabling you to see the relationships among metadata elements in the catalog.

➤ To view the lineage of a metadata element:

1 In the Metadata Navigator, right-click a metadata element and select Show Lineage.
2 The lineage for the element is displayed in the lineage work area, the middle pane of the Essbase Studio Console.

For information on optional tasks you can perform in the lineage work area, see “Using the Lineage Work Area” on page 226

Using the Lineage Work Area

The lineage work area allows you to perform the following tasks:

- **Navigate through the lineage**— Click the thumbnail viewer icon, in the bottom right of the lineage work area. The thumbnail viewer in the lower right corner of the lineage work area contains the same portion of the lineage in the main lineage work area, in a miniature format. A smaller, transparent blue pointer covers a portion of the lineage diagram. Use the thumbnail viewer to navigate in a thumbnail view of your lineage to the point on which you want to focus.

- **View the history of an element**— Place your cursor over an element, and a tool-tip shows a brief history of the element, including the date it was created, who created it, and the date on which it was modified.

  Note:
  Holding your cursor over elements causes concatenated names to fully display.

- **Focus on one element**— To see the lineage of only one element, double-click the element.

- **Move element**— To move an element, left-click it and drag it to the desired location.

- **Sweep and move multiple elements**— To move several elements at once, left-click and sweep over the appropriate elements. Then left-click any one of the selected elements and drag it to the desired location. All the selected elements will move to that location.

- **Undo**— To undo an element move, right-click in the lineage work area and select Undo.

- **Redo**— To redo an element move, right-click in the lineage work area and select Redo.

- **Zoom In**— To get a larger view of the lineage, right-click in the lineage work area and select Zoom In.

- **Zoom Out**— To get a smaller view of the lineage, right-click in the lineage work area and select Zoom Out.

- **Launch the Administration Services Console**— Right-click in the lineage work area and select Launch Essbase Administration Services Console.
- **Edit properties**— Right-click on the element and select Properties.
  The dialog box for editing the element is displayed.

- **Edit Essbase properties**— To edit the properties of an Essbase model, right-click on the model and select Essbase Properties.
  The dialog box for editing the element is displayed.

- **Show deployment history**— To see the deployment history of a cube schema, right-click in the cube schema and select Show Deployment History.

- **View sample data**— To see the sample data for a member, right-click on the member and select View Sample Data.
Drill-through Reports Overview

With drill-through reports, you create spreadsheet reports that display data retrieved directly from external related data sources.

When you create an Essbase database, you do not use all of the data of the associated external data source. Instead, you choose the data that spreadsheet users are most likely to need and summarize it in a way that is useful to them. For example, you might summarize the amount of root beer sold in each state. The summaries enable spreadsheet users to compare sales across states and regions. They can answer questions such as "Which state sold the most root beer in January?" and "Which state sold the most root beer in July?"

Some spreadsheet users may need more detailed information than the summaries can provide to create a strategy to increase root beer sales. They need to answer the following types of questions:

- What are the ten lowest performing stores in California?
- What are the addresses of the stores?
- What are the names of the store managers?

Detail information such as this is usually not stored in an Essbase database, but it is stored in an external data source.

To view the relevant detail data, spreadsheet users drill through to the external data source and view a report containing the data in the context of the data in the Essbase database. This report is a drill-through report.

Until release 11.1.1, you created drill-through reports using Integration Services. Starting with release 11.1.1, you can also create drill-through reports in Essbase Studio. Essbase Studio expands and enhances the scope and capabilities of drill-through reports.
A drill-through report can be based on any intersection context or level (member combination) defined in the selected data source. In Smart View, visual clues, such as font color or size, alert the user that a drill-through report is available on one or more cells in the spreadsheet. In Oracle Essbase Spreadsheet Add-in, cubes built using Essbase Studio cannot provide those visual clues for drill-through cells so any cell is a potential drill-through cell. To learn how to access a drill-through report in Spreadsheet Add-in while connected to a cube built by Essbase Studio see “Guidelines for Accessing Drill-through Reports in Spreadsheet Add-in” on page 268.

In Smart View, users access drill-through reports by selecting Ad Hoc Analysis, then Drill-Through Reports from the Hyperion menu; in Spreadsheet Add-in, users double-click intersections to access the Linked Object Browser to start the drill-through process.

**Note:**

In Spreadsheet Add-in, to access drill-through reports from cubes built through Essbase Studio, users must be registered with Shared Services and be provisioned with, at a minimum, the Essbase Read role and the Essbase Studio cpViewer role.

**Note:**

When logging in to Essbase in Spreadsheet Add-in, login names and password are case-sensitive for users who are provisioned through Shared Services.

Spreadsheet users view reports from predefined drill-through targets that retrieve the relevant detail from the data source. Drill-through report targets created in Essbase Studio include:

- Relational databases
- URLs, including FDM and OBI URLs
- Customized Java methods

Drill-through report targets created in Integration Services include relational databases and URLs.

This chapter contains information on creating and maintaining drill-through reports in Essbase Studio. For more information about:

- Creating and maintaining drill-through reports in Integration Services, see the *Oracle Essbase Integration Services Online Help*.
- Using drill-through reports with Smart View, see the *Oracle Hyperion Smart View for Office User’s Guide* and *Oracle Hyperion Smart View for Office Online Help*.
- Using drill-through reports with Essbase Spreadsheet Add-in, see the *Oracle Essbase Spreadsheet Add-in User’s Guide* and *Oracle Essbase Spreadsheet Add-in Online Help*.

## Working with Drill-through Reports

Drill-through reports provide spreadsheet client users direct access to data stored in a related physical data source. Drill-through reports are based on intersection levels (member sets). In a
spreadsheet, users double-click the cells representing these intersection levels to view detail information that is stored in the target data source, not in the Essbase cube.

These topics describe the steps involved in defining a drill-through report:

- “Creating a Drill-through Report” on page 231
- “Specifying Report Intersection Levels” on page 231
- “Defining the Report Type and Customizing the Report” on page 233
- “Associating Drill-through Reports with Essbase Models” on page 245

Creating a Drill-through Report

To create a drill-through report:

1. In the Metadata Navigator, select the folder where you want to store the drill-through report, right-click and select New > Drill-through Report.

   If you right-click on a folder when you select the Drill-through Report option, the drill-through report is stored in that folder; if you right-click on Root, the drill-through report is stored at the root level.

2. Enter a Name for the drill-through report.

   Drill-through report names are limited to 50 characters.

   Drill-through report naming limitation are described in “Drill-through Reports Limitations” on page 268

3. Optional: Enter a Description for the drill-through report.

4. Complete these tasks:

   - Specify the report intersection levels; see “Specifying Report Intersection Levels” on page 231
   - Define the report type and customize the report; see “Defining the Report Type and Customizing the Report” on page 233
   - Associate the report with one or more exported models; see “Associating Drill-through Reports with Essbase Models” on page 245

Specifying Report Intersection Levels

In the Context tab of the drill-through report editor, when you define one or more intersection levels for a drill-through report, you set the common thread that you want this drill-through report to have with other drill-through targets.

See “Determining Where to Set an Intersection Level” on page 232 for more information.
To specify report intersection levels:

1. **In the Context tab of the drill-through report editor, click the Add button.**
   The Select Hierarchy dialog box is displayed.

2. **In Select Hierarchy, navigate to a hierarchy that you want to include in the drill-through report, select it, and click OK.**
   The hierarchy is displayed under Intersections in the drill-through report editor.

   **Note:**
   All hierarchies must be selected from the same relational source. You may add one or more hierarchies for all drill-through intersections you want to define in this report.

   **Note:**
   Hierarchies created from Performance Management Architect sources are not supported as drill-through intersections.

   You may also add hierarchies by dragging them from the Metadata Navigator directly to the area under Intersections in the Context tab of the drill-through report editor. When you use this method, you do not need to click the Add button.

3. **In the expanded hierarchy under Intersections, select the check box next to the member name which you want to specify as a drill-through intersection.**

   **Note:**
   You can specify any level in the hierarchy, including the top level, as a drill-through intersection.

4. **Repeat steps step 2 through step 3 to add hierarchies for all drill-through intersections you want to define in this report.**

   **Note:**
   In Smart View only: If you select only one hierarchy, then the drill-through report will be available from the corresponding member cells as well as from the data cells. This is known as “member drill-through.”

5. **Select the Report Contents tab to define the report type and specify the accompanying parameters, described in “Defining the Report Type and Customizing the Report” on page 233.**

---

**Determining Where to Set an Intersection Level**

When you specify an intersection, or context, in a drill-through report, you determine the Essbase cells from which a spreadsheet user can drill through.

If you specify the drill-through context to contain the top level of a hierarchy, spreadsheet users can drill through from any intersection that contains the related Essbase dimension. For
example, if you set an intersection level for the entire Product hierarchy, spreadsheet users can drill through from any level in the Product dimension that was build from that Product hierarchy. In the spreadsheet, users can double-click any data value that involves a member of the Product dimension, such as Colas or 100-10.

Specifying a level as a drill-through context will ensure that a filter (or a WHERE clause in the SQL SELECT statement) based on that level in the hierarchy is applied to the drill-through request.

Defining the Report Type and Customizing the Report

In the Report Contents tab, you define the type of report you want to create. The tasks you must complete depend on the type of report you define.

The report types you can choose are:

- Relational—select if the report uses a relational query. The relational query can be either standard SQL generated by Essbase Studio or user-defined Template SQL.
  
  See “Defining and Customizing a Report for a Relational Source” on page 233.

- URL—select if the report points to a URL, which is launched in a browser.
  
  See “Defining and Customizing a Report to a URL” on page 238.

- Java Method—select if the report points to a user-defined Java method.
  
  See “Defining and Customizing a Report to a Java Method” on page 243.

Defining and Customizing a Report for a Relational Source

Use the Report Contents tab to perform the following tasks:

- Define the report type and specify the columns and their order; see “Defining the Relational Report Type and Specifying Columns and Their Order” on page 233

- Define Template SQL, see “Defining Template SQL” on page 235

- Specify row governors—see “Specifying Row Governors for Relational Sources” on page 237

- Specify filters, see “Specifying Drill-through Report Filters for Relational Sources” on page 237

- Test the report, see “Testing Reports for Relational Sources” on page 238

Defining the Relational Report Type and Specifying Columns and Their Order

➤ To define a relational drill-through report type and specify the report columns and their order:

1 In the Report Contents tab of the drill-through report editor, select Relational from Drill-through Report Type.
2 Add the columns that will appear in the drill-through report.

**Note:**
All columns in the report must be from the same relational source.

a. Click **Add** next to the report grid of the **Report Contents** tab to add a drill-through column to the report.

A drill-through column is the external database column that Essbase Studio retrieves when a spreadsheet user double-clicks an intersection level. Columns contain detail information that is not available in the Essbase database; for example, a list of store managers.

b. In **Select Column**, navigate to the column you want to include in the drill-through report, select it, and click **OK**.

The column you selected is displayed in the report grid.

c. Repeat step 2.a through step 2.b for each column you want to add to the drill-through report.

3 **Optional:** To display a column name other than the name of the column that you selected in step 2:

a. In the report grid, select the column to work with and then click in the cell under the **Display Name** column heading to activate the cell.

b. Type the column name that you want to display in the drill-through report.

c. Repeat step 3.a through step 3.b for each column for which you want to change the display name in the drill-through report.

4 **Optional:** To apply an aggregate function to a drill-through column:

a. In the report grid, select the column to work with and then click in the cell under the **Aggregate** column heading to activate the drop-down list control.

b. Select one of the following aggregate functions from the drop-down list:

   - **Avg**—Returns the average value of the column. Applies only to numeric column types.
   - **Count**—Returns the number of selected rows. Applies to all column types.
   - **Min**—Returns the minimum value of the column. Applies only to numeric column types.
   - **Max**—Returns the maximum value of the column. Applies only to numeric column types.
   - **Sum**—Returns the total sum of the column. Applies only to numeric column types.

c. Repeat step 4.a through step 4.b for each column to which you want to apply an aggregate function.

5 **Optional:** To define the sort order of a drill-through column:

a. In the report grid, select the column to work with and then click in the cell under the **Sort Order** column heading to activate the drop-down list control.

b. Choose the Asc (ascending order) or Desc (descending order) option.
Repeat step 5.a through step 5.b for each column to which you want to apply a sort order.

6 Use the Move Up and Move Down buttons to arrange the order of the columns in the drill-through report.

Columns are displayed in a report from left to right in the order in which they are listed from top to bottom in the report grid. Arrange the columns in the report grid in the order in which you want them to be displayed in the drill-through report.

See “Row Sort Order Overview” on page 235 for more information.

7 Optional: To delete columns from the drill-through report, select the column in the report grid and click Delete.

8 Optional: To display duplicate records in the drill-through report, select the Show duplicate records check box.

9 Optional: If you want to base this report on SQL that you have written, select the Use user-defined SQL check box, and click the Template SQL button.

See “Defining Template SQL” on page 235 for more information.

Alternatively, if you have saved user-defined SQL in the Template SQL dialog box, but do not want to use it, clear the “Use user-defined SQL” check box. Essbase Studio will automatically generate the SQL that is used to create a drill-through report, and your drill-through SQL is saved for future use.

10 To complete the other remaining tasks in the Report Contents tab of the drill-through report editor, see the following topics:
   ● “Defining Template SQL” on page 235
   ● “Specifying Row Governors for Relational Sources” on page 237
   ● “Specifying Drill-through Report Filters for Relational Sources” on page 237
   ● “Testing Reports for Relational Sources” on page 238

Row Sort Order Overview

You can determine the order in which the spreadsheet client displays the rows and the contents of the rows it retrieves; for example, you can sort the contents of the PRODUCTDIM.SKUNAME column in descending order (from highest to lowest value). This sort presents the products in reverse alphabetical order, from Vanilla Cream to Old Fashioned to Caffeine Free Cola.

If you do not determine the sort order, spreadsheet users view data in the order determined by the external data source. SQL Server, for example, sorts members in ascending order by default.

Defining Template SQL

For information on Template SQL, see “Template SQL Overview” on page 236.

Note:

This procedure is optional.
To define Template SQL:

1. From the Report Contents tab of the drill-through report editor, click the Template SQL button.

2. Optional: To use standard SQL as a reference, from the Cube Schema drop-down list, select the cube schema on which you want to base your SQL, and click Get Standard SQL.

   The SQL for this cube schema is displayed in both the read-only Standard SQL text box, and the editable User-defined SQL text box.

3. Edit or write the SQL you require for this drill-through report.

4. Click Update User-defined SQL to save your user-defined SQL without exiting the dialog box.

5. Click Validate to validate the syntax in User-defined SQL.

6. Note any errors and make the appropriate corrections to the user-defined SQL.

   **Note:**

   When the SQL is correct, a message displays telling you the SQL is valid.

7. When any errors are corrected, click OK to return to the drill-through report editor.

8. To complete the other remaining tasks in the Report Contents tab of the drill-through report editor, see the following topics:
   - “Specifying Row Governors for Relational Sources” on page 237
   - “Specifying Drill-through Report Filters for Relational Sources” on page 237
   - “Testing Reports for Relational Sources” on page 238

**Template SQL Overview**

Essbase Studio automatically generates the SQL that is used to create a drill-through report. The SQL statement identifies the OLAP intersection levels of the drill-through report and the columns being returned. You can override the SQL generated by Essbase Studio with your own SQL.

Using the Template SQL dialog box, you define a template for drill-through SQL, referred to as "template SQL," that specifies which parameters from the current reporting context need to be passed to the drill-through report as parameters. Template SQL can incorporate tables and columns from any defined data source connection, whether or not it is used or joined in a minischema.

Essbase Studio Template SQL uses the "$\$" syntax as the variable delimiter. All text contained within the "$\$" syntax is replaced with actual column or data values during drill-through execution. However, you cannot use the $$ substitution variables in template SQL when the intersection level of the dimension is defined at Generation 1 and the dimension is built from a parent/child table.
Specifying Row Governors for Relational Sources


**Note:**
This procedure is optional.

To specify a row governor, enter the maximum number of rows to retrieve in the **Row governor** text box.

For example, to stop a query that retrieves more than 200 rows, enter **200**.

To complete the other remaining tasks in the Report Contents tab of the drill-through report editor, see the following topics:

- “Defining Template SQL” on page 235
- “Specifying Drill-through Report Filters for Relational Sources” on page 237
- “Testing Reports for Relational Sources” on page 238

Specifying Drill-through Report Filters for Relational Sources

Specify a filter to limit the results returned for specific target drill-through columns.

To specify drill-through report filters for relational sources:

1. In the **Report Contents** tab of the drill-through reports editor, click **Filter**.
2. In the **Drill-through Filter** dialog box, drag the appropriate elements from the lists of elements in the **Source**, **Functions**, and **Operators** tabs and drop them in the **Filter** box.

For example, to filter a drill through report for members in the “400” product family, you may enter a filter such as the following:

```sql
'contains'( connection : '\tbcSource'::'tbc.family'.'FAMILY'="400")
```

To filter a drill-through report for members of the “400” product family in the state of New York only, you may enter a filter such as the following:

```sql
contains(connection : '\tbcSource'::'tbc.market'.'STATE',"New York") and contains(connection : '\tbcSource'::'tbc.family'.'FAMILY'","400")
```

3. Click **OK** to close the **Drill-through Filter** dialog box and return to the **Report Contents** tab of the drill-through report editor.
4. To complete the other remaining tasks in the **Report Contents** tab of the drill-through report editor, see the following topics:
   - “Defining Template SQL” on page 235
   - “Specifying Row Governors for Relational Sources” on page 237
Testing Reports for Relational Sources

Use this procedure for testing drill-through reports based on relational sources.

➤ To test the drill-through report:

1 In the Report Contents tab, click Test.

2 Select a Cube Schema from the drop-down list to use for testing.

3 Provide a Column value for each intersection.

Under Intersection, the columns you selected as intersections on the Context tab are displayed. You may also click in the intersection cell, select the down arrow at the right of the cell, and choose a column from a drop-down list of all available columns under the same intersection. For each intersection you select, enter a valid value under Column Value.

For example, using the TBC sample database, if Family is displayed as an intersection, enter 300 in the Column Value column to view results for the 300 product family. Further, use the drop-down list in the intersection cell for Family to select another column from a list of available drill-through columns; for example, SKU, and enter 300-30 in the Column Value column.

4 Optional: Enter an integer to specify the Maximum rows to display.

The default is to display 20 rows.

5 Click Show Result.

View the Resulting DT Report section to see how results will be displayed in the drill-through report.

6 Click Close to return to the Report Contents tab of the drill-through report editor.

7 Click Save and then Close to close the Drill-through Report Properties dialog box or click the Associations tab and perform the tasks in “Associating Drill-through Reports with Essbase Models” on page 245.

Defining and Customizing a Report to a URL

 Essbase Studio expands your options for drill-through reports by providing the capability of specifying a URL as a drill-through target. When you specify a URL in the Report Contents tab of the drill-through report editor, users are able to drill through directly to a URL from a drill-through intersection in a spreadsheet.

Use the URL report type and then define the target URL. You can specify static URLs as drill-through targets, as well as dynamic URLs to targets such as Oracle Hyperion Financial Data Quality Management, Fusion Edition and Oracle Business Intelligence (OBI) web pages.

The URL syntax must be consistent with the requirements of the target URL. However, to express dimensions, columns, and values, all URLs have the following variable structure in common:

$$<dimension-name>-VALUE$$
When a user clicks on a drill-through cell, Essbase Studio makes the substitutions necessary to generate the target URL in the context of the selected drill-through intersection.

For example, if the Market dimension is in the point-of-view (POV) of a drill-through target URL, the variable for the value from Market will look like this:

$$Market-VALUE$$

When the spreadsheet user selects the drill-through cell from an intersection containing a member from the Market dimension, Essbase Studio substitutes the appropriate Market value into the URL syntax; for example:

East

In the following example, the variable for the value from the Product dimension will look like this:

$$Product-VALUE$$

When the spreadsheet user selects the drill-through cell from an intersection containing a member from the Product dimension, Essbase Studio substitutes the appropriate Product value in the URL syntax; for example:

Cola

Note:

Drill-through target URLs cannot exceed 256 characters in length.

To define a URL report type and specify the URL to drill-through to:

1. In the Context tab of the drill-through report editor, select URL from Drill-through Report Type.
2. In the URL text box, enter the target URL for the drill-through report.
   
   Alternatively, you may create the URL by editing the syntax generated by Essbase Studio when you click one of the following buttons:

   - **Sample URL Template**—Provides the syntax for a target URL based on the intersections you specified in the Context tab.
     
     See “Sample URL Template” on page 240 for more information.
   
   - **FDM URL Template**—Provides the syntax for a target FDM URL based on the intersections you specified in the Context tab.
     
     See “Sample FDM URL Template” on page 240 for more information.
   
   - **OBI URL Template**—Provides the syntax for a target OBI URL based on the intersections you specified in the Context tab.
     
     See “Sample OBI URL Template” on page 242 for more information.

3. Click Save; then Close to close the drill-through report editor, or select the Associations tab and perform the tasks in “Associating Drill-through Reports with Essbase Models” on page 245.
Sample URL Template

Use the Sample URL Template to help you define a URL as a drill-through report target. You can specify static HTTP URLs as drill-through targets, as well as dynamic URLs that take into account the drill-through intersections specified in the Context tab of the drill-through report editor.

Note:
Drill-through target URLs cannot exceed 256 characters in length.

Syntax
You may provide a static HTTP URL; for example:
http://www.oracle.com

For a URL to a dynamic target, Essbase Studio Console provides a sample that uses Google. The number of values to search on depends on the number of drill-through intersections specified in the Context tab of the drill-through report editor. The following is the syntax when drill-through intersections were specified on two hierarchies, Product and Market:
http://www.google.com/search?hl=en&q=$$Product-VALUE$$+$$Market-VALUE$$

Essbase Studio makes the appropriate substitutions to the URL, depending on the drill-through intersection context, when the spreadsheet user executes the drill-through report.

Example
The example uses the sample URL template from the Essbase Studio drill-through report editor and assumes that the Product and Market hierarchies were selected in the Context tab of the drill-through report editor.

When the user clicks on a drill-through cell in a Product and Market intersection, Essbase Studio makes substitutions for the variables $$Product-VALUE$$ and $$Market-VALUE$$, and generates the following URL string:
http://www.google.com/search?hl=en&q=Cola+East

Sample FDM URL Template

The Sample FDM URL Template provides the syntax for a target FDM URL based on the intersections you specified in the Context tab of the drill-through report editor.

The syntax and examples assume that the Product and Market hierarchies are selected in the Context tab of the drill-through report editor.

Syntax
http://<server-name>/HyperionFDM/AuthorizedPages/
IntersectionSummaryByLocation.aspx?FDMTargetProdID=<prod-id>&FDMadapterKey=<adapter-key>&FDMTargetAppName=<target-app-
Note:
Drill-through target URLs cannot exceed 256 characters in length.

Parameters

- `<server-name>`—The name or IP address of the server to which you want to connect.
- `<prod-id>`—The FDM target product ID. This is the product that is drilling through to FDM so that FDM can do the dimension mapping. The value supplied for this named parameter should be an arbitrary string that uniquely identifies the consolidation/analysis product; for example, HFM, Planning, HSF, Enterprise, or Essbase.
- `<adapter-key>`—The FDM adapter key that is being used to connect to the target system. An example of the field would be ES9x-G4-A for Essbase. Note that this parameter needs to be able to be updated as users will update Adapters with new releases of FDM.
- `<target-app-name>`—The FDM application name. This value should match the FDM administrator-assigned identifier given to the particular data set or repository within the product. For example, for the Comma sample application that ships with Oracle Hyperion Financial Management, Fusion Edition, the value would be comma.
- `Product=```Product-VALUE```$—The Point-Of-View (POV) dimension and the dimension member value, in this case, a value from the Product dimension. Essbase Studio automatically generates the dimension name. Upon execution of the drill-through report, a value (for example, Cola) is substituted in the URL so that `Product=```Product-VALUE```$ becomes:
  
  Product=Cola

If a dimension is called by a different name in the FDM source, you must make that substitution manually in the drill-through syntax. For example, if the “Product” dimension is called the “Items” dimension in FDM, you must substitute Items for the dimension name in the Essbase Studio-generated syntax, as follows:

http://<server-name>/HyperionFDM/AuthorizedPages/IntersectionSummaryByLocation.aspx?FDMTargetProdID=<prod-id>&FDMadapterKey=<adapter-key>&FDMTargetAppName=<target-app-name>&SSO_TOKEN=$$CSS-TOKEN$$&FDMAppName=<app-name>&Items=```Product-VALUE```$&Market=```Market-VALUE```$

Note that all visible dimensions in a target FDM report must be specified in the FDM URL syntax, even if, for example, you are only interested in the information in one dimension in the report.

By default, the Sample FDM Template syntax picks up all intersections you specified in the Context tab of the drill-through report editor and adds them to the template syntax statement. If there are any remaining dimensions in the target FDM report, you must also include those in your syntax statement.
Example

The following is the URL template syntax for a drill-through report that drills through to a target FDM report containing the dimensions Scenario, Year, Accounts, Market, and Product.

The Scenario, Year, and Product dimensions were specified as intersections in the Context tab of the drill-through report editor. Essbase Studio automatically adds these to the FDM template syntax.

The Accounts and Market dimensions are part of the Oracle Hyperion Financial Data Quality Management, Fusion Edition target report, but were not specified as intersections in our drill-through report. Therefore, you must explicitly specify the dimension and member names in the URL syntax, as shown next:

http://oak3/HyperionFDM/AuthorizedPages/IntersectionSummaryByLocation.aspx?
FDMTargetProdID=Essbase&FDMadapterKey=ES9x–G4–
A&FDMTargetAppName=Demo&SSO_TOKEN=%%CSS-TOKEN%%&FDMAppName=TEST95&Scenario= $$$Scenario-VALUE$$&Year=%%Year-VALUE%%&Product=%%Product-VALUE%%&Accounts=Sales&Market=Nebraska

Sample OBI URL Template

The Sample OBI URL Template provides the syntax for a target FDM URL based on the intersections you specified in the Context tab of the drill-through report editor.

The syntax and examples assume that the Product and Market hierarchies are selected in the Context tab of the drill-through report editor.

Syntax

After selecting the Product and Market hierarchies in the Context tab of the drill-through report editor:

obi:http://<server-name>:9704/analytics/saw.dll?

Note:

Drill-through target URLs cannot exceed 256 characters in length.

Parameters

- `<server-name>`—The name or IP address of the server to which you want to connect.
- `<number-of-parameters>`—Number of report parameters.
- `<Market-column>` and `<Product-column>`—The column in the OBI data source to retrieve in the target drill-through report. For example, in the Product dimension, you may specify the column, SKU; in the Market dimension, you may specify the column, Region.
Defining and Customizing a Report to a Java Method

Use the Java Method report type to define a drill-through report to drill through to a custom Java method that you create.

You may create a Java method that generates a report that satisfies your specific requirements. For example, you may create a Java method that extracts data from a specific application.

For the Spreadsheet Add-in or Smart View user in the client application that runs the drill-through report, there is no difference between executing a drill-through report to a relational source and executing a drill-through report with user-defined Java methods.

➤ To define a Java method report type and specify the Java method to execute during drill-through:

1. From Drill-through Report Type, select Java Method.
2. In Java Class Name, enter the full package name of the Java class; for example:
   test.com.hyperion.cp.scripts.acceptance.test_cases.TestDTRJavaMethod
3. In Java Method Name, enter the name of the Java method; for example: runTest

User-defined Java methods that can be invoked by Essbase Studio Server must have the following signature:

```
public static void <method-name>(ArrayList<String>[] args, ResultSet result)
```

The `ArrayList<String>[]` `args` parameter of the user-defined Java method is an array list of drill-through report arguments. Each item of the array list describes a member from the intersection and includes the dimension name, class name, and member name. For example:

```
"ProductH"   "SKU"         "100-10"
"TimeH"      "Time Year"   "2006"
```

If the member is a top member of an Essbase dimension, the value of class name is null. For example:

```
"ProductH"   null     "ProductH"
```
The ResultSet parameter is an interface from the Java package, com.hyperion.cp.interfaces.

**Note:**
Before setting any values in the records of the result set, all tags of the result set must be defined.

The following methods of the interface can be used in user-defined Java methods:

```java
/**
 * Adds a new tag to result set signature
 * @param tag Tag name
 * @param clazz Java class that corresponds to the tag
 *              (Boolean, Integer, Long, Double, String)
 */
public void addTag(String tag, Class clazz);

/**
 * Sets value in result set record
 * @param tag Tag name
 * @param value Value
 */
public void setValue(String tag, boolean value);
public void setValue(String tag, int value);
public void setValue(String tag, long value);
public void setValue(String tag, double value);
public void setValue(String tag, String value);

/**
 * Adds record to the result set
 */
public void addRecord();
```

The following is an example of the user-defined Java method, runTest. This method converts the parameters of the drill-through report to a result set with the tags Hierarchy, Class, Value. The values of all tags are strings.

```java
package test.com.hyperion.cp.scripts.acceptance.test_cases;

import com.hyperion.cp.interfaces.ResultSet;
import java.util.ArrayList;
import java.util.Iterator;

public class TestDTRJavaMethod
{
    /**
     * User defined java method for DTR
     * @param result Result of DTR
     * @param args Actual arguments of the DTR
     */
    public static void runTest(ArrayList<String[]> args, ResultSet result)
    {
        // make signature of the result
        result.addTag("Hierarchy", String.class);
```
When the spreadsheet user runs the drill-through report, the Java method is executed and results in the following drill-through report:

```
<table>
<thead>
<tr>
<th>Hierarchy</th>
<th>Class</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProductH</td>
<td>SKU</td>
<td>100-10</td>
</tr>
<tr>
<td>TimeH</td>
<td>Time Year</td>
<td>2006</td>
</tr>
<tr>
<td>MarketH</td>
<td>REGION</td>
<td>East</td>
</tr>
<tr>
<td>MyMeasuresH</td>
<td>Msr_Gen4</td>
<td>Sales</td>
</tr>
</tbody>
</table>
```

4 Click Save; then Close to close the drill-through report editor, or select the Associations tab and perform the tasks in “Associating Drill-through Reports with Essbase Models” on page 245.

### Associating Drill-through Reports with Essbase Models

You can associate a drill-through report with any or all models in the catalog that contain the same intersections defined in the report.

➤ To associate a drill-through report with an Essbase model:

1 **In the Associations tab of the Drill-through Report Properties dialog box**, select the check box next to the Essbase model names that you want to associate with this drill-through report.

To associate all models in the catalog that contain the intersections defined in the current drill-through report, select the “Select all Essbase models” check box.

If you make no selections in the Associations tab:

- In Smart View, no cells will be marked as drill-through cells in the spreadsheet.
- In Essbase Spreadsheet Add-in, all data cells will behave as drill-through cells, even though no drill-through reports are present. See “Guidelines for Accessing Drill-through Reports in Spreadsheet Add-in” on page 268 for information about this behavior.
2 Optional: To view the most up-to-date list of Essbase models containing the intersections defined in the current drill-through report, click the Refresh model list button.

3 Click Save; then Close to close the drill-through report editor.
Find and Search

Overview

Essbase Studio offers two methods to help you locate metadata elements, Find and Search. Find focuses on looking for metadata within selected UI components of the Essbase Studio Console. For example, you may choose to find a metadata element in the Metadata Navigator, the Minischema pane, the Data Sources tab or Minischema tab in the Physical Navigator. Further, you may select a folder in the Metadata Navigator or Physical Navigator and perform the Find only within that folder.

Search performs searches on your entire metadata catalog. Alternatively, you can narrow the search to look only for specific types of metadata elements, such as hierarchies or cube schemas.

For more information, see “Finding Metadata Elements” on page 247 and “Searching for Metadata Elements” on page 248.

Finding Metadata Elements

Use Find to locate a metadata element within a particular UI component, such as the Metadata Navigator, the Minischema pane, the Data Sources tab or Minischema tab in the Physical Navigator.

➤ To find a metadata element:

1. **Place the cursor within the Essbase Studio Console UI component in which you want to perform a find.**

   For example, to find a metadata element within the graphical view of the minischema, open the minischema by right-clicking on it in the Minischema tab of the Physical Navigator and select Edit.

2. **Select Edit > Find to display the Find dialog box.**

3. **In the Find text box, enter the find string.**
For example, to locate tables or columns in the minischema containing the string, product, type product in the Find text box.

4 **Optional—Select one or more options to refine the Find:**

- **Match Case**—Find only those metadata elements which match your find string and have the exact same upper and lower case.
  
  For example, if you enter Product as your find string and select Match Case, the find locates only metadata elements called “Product;” it will not locate metadata elements called “PRODUCT.”

- **Match Word**—Find only those elements which match your find string and are whole words.
  
  For example, if you enter product as your find string and select Match Word, the find locates only metadata elements called “product;” it will not locate metadata elements called “productdim.”

- **Apply to current selection**—Find only those elements within a certain on-screen selection.
  
  For example, if you select one or more folders in the Metadata Navigator, enter a find string, and then select the “Apply to current selection” option, the find is performed only within the selected folders.

5 **Click the Find button.**

Within the UI element you selected, the find stops on the first occurrence of your find string and highlights it in yellow. For example, if you performed a find on an entire minischema in the minischema pane, the find stops on the first occurrence of the find string within the minischema, whether it is a minischema table or column.

6 **Click Find again to locate the next occurrence of the find string.**

7 **Repeat step 6 until the message “No more occurrences” is displayed at the bottom of the Find dialog box.**

---

**Searching for Metadata Elements**

Use Search to locate metadata elements in your catalog. This is useful when your library of metadata elements is very large.

➢ To search for metadata elements:

1 **Select Edit > Search Metadata Elements.**

   The search interface is launched as a tab in the left pane of the Essbase Studio Console.

2 **Enter the Name of the metadata element to search for.**

3 **In Type, select <All> or use the drop-down to narrow your search to a specific type of metadata element.**

   You can perform searches on the following metadata elements:

   - Hierarchies
   - Folders
   - Essbase models
● Metadata Elements
● Drill-through reports
● Cube schemas

4 Optional—To narrow the search location:
   a. In Look In, click Browse and, in the Search Folder dialog box, navigate to the folder you want to search.
   b. Double-click the folder to be searched to close the dialog.

The entire content of the Metadata Navigator is searched by default.

5 Click Search, and then review Search Results.

6 Optional—To clear the search criteria and the results, and begin a new search, click Clear, then repeat step 2 through step 5.

7 To close the Search dialog box, click the X in the Search tab.

Note:
The metadata element names you enter in the Name field of the Search dialog box are retained and can be selected from the drop-down list in future searches.
In This Chapter

Setting General Preferences .................................................................................................... 251
Setting Schema Preferences....................................................................................................2 52

Setting General Preferences

Set general preferences regarding the messages and warnings you want displayed in Essbase Studio Console.

➢ To set general preferences:

1 Select Edit > Preferences.

2 To display a confirmation message when deleting items from the Source Navigator, Metadata Navigator, or work area, select the Show Confirm Delete dialog check box.

3 To control the types of messages that display in the Console Message pane, select or clear the following options in the Console Messages group:
   ● Show informational messages
   ● Show warning messages
   ● Show error messages

4 To direct the Help > Help Topics menu item to the location of the online help, click Browse, and browse to the folder location where you placed the online help.

   The help file location is also used to call the help when you press the F1 key in dialog boxes and console panes.

   Tip:
   
   Place the online help in a folder called “help” in a location easily accessible by Essbase Studio Console; for example, in the <essbase_studio>\help folder.

5 Click Apply to apply your selections, then click OK.

   Note:
   
   To return to the default preference settings, click Restore Defaults.
Setting Schema Preferences

Set schema preferences to control the display of minischemas in Essbase Studio Console.

➤ To set schema preferences:

1 Select Edit > Preferences.

2 To control the display of data source objects in the table objects of a minischema, select or clear the following options in the Relational table display options
   - To display symbols representing columns types to the left of the column name, select Column type indicator
   - To display a key symbol to the right of the primary key column name select Primary Key
   - To display a symbol representing columns that have foreign key relationships, select Foreign Key

3 To control the way join lines are displayed between joined tables, select an option from the Join Connector group:
   - Diagonal—allows join lines to be drawn diagonally between tables, where appropriate
   - Rectangle—allows join lines to be drawn only at right or left angles between tables, where appropriate

4 To control the display of the ends of join lines between tables, select an option from the Join Ends group:
   - Arrow—displays an arrow at the end point of the join
   - Crow’s Feet—displays a three-pronged “crow’s feet” pattern at the end point of the join

5 Click Apply to apply your selections, then click OK.

Note:

To return to the default preference settings, click Restore Defaults.
Catalog Migration Overview

Note:

The procedures described in this overview are addressed to database administrators and those who have knowledge of relational database concepts, Integration Services, and the specific relational databases in their working environment.

Integration Services stores metadata in a relational repository called the catalog. Essbase Studio also has a catalog containing metadata. Essbase Studio provides a mechanism to migrate the metadata in your Integration Services catalog to the Essbase Studio catalog.

The following topics are discussed in this overview:

- “Model Migration” on page 253
- “Metaoutline Migration” on page 254

Model Migration

Essbase Studio performs several tasks when migrating a model from an Integration Services catalog to an Essbase Studio catalog.

The Essbase Studio migration utility first creates a folder with the same name as the model in Integration Services, and then it performs the migration.

An Integration Services model contains a star schema, which contains a subset of tables, as well as information about star schema joins. If a user has defined hierarchies, that information is also in the model.

To migrate this metadata to Essbase Studio, the migration utility performs the following steps:
A subfolder is created under the model folder in Essbase Studio for each of the tables in the star schema.

For each table column, a dimension element is created within each table folder. The caption and key binding for each dimension element reflects the table column.

For time dimension elements, the appropriate time transformations are applied and new dimension elements are created.

For each hierarchy defined in the Integration Services model, a corresponding hierarchy is created in the Essbase Studio model folder.

A minischema is created in Essbase Studio corresponding to the star schema of the Integration Services model so that all joins are migrated, including the self-joins for recursive hierarchies.

**Metaoutline Migration**

Essbase Studio performs several tasks when migrating a metaoutline from an Integration Services catalog to an Essbase Studio catalog.

The Essbase Studio migration utility first creates a subfolder (in the model folder) with the same name as the metaoutline in Integration Services. The migration utility then performs the actual migration.

The following topics are discussed in this section:

- “Mapping Hierarchies” on page 254
- “Mapping the Measure Column” on page 254
- “Creating the Essbase Export Model” on page 255
- “Mapping Aliases, User-defined Members, and Reports” on page 255

**Mapping Hierarchies**

An Integration Services metaoutline consists of dimensions and attribute dimensions. Each dimension can have alternate hierarchies. The migration utility maps the Integration Services dimension to the Essbase Studio hierarchy as follows:

- For each Integration Services dimension, a corresponding hierarchy is created.
- Each attribute dimension is created as an alternate hierarchy in Essbase Studio.
- For each Integration Services recursive dimension, a hierarchy is created by identifying parent and child members.
- If the Integration Services metaoutline contains an accounts dimension that is not based on the fact table, an accounts dimension will not be created from the fact table in Essbase Studio.

**Mapping the Measure Column**

After the hierarchies are mapped, the migration utility creates a cube schema. The measure column is then mapped according to one of two scenarios:
● If the Integration Services metaoutline contains an accounts dimension that is based on the fact table, a measure hierarchy is created and added to the cube schema.

● If the Integration Services metaoutline contains an accounts dimension that is not based on the fact table, the measure column is added as a loose measure.

Creating the Essbase Export Model

After the cube schema is created and the measure column is mapped, the migration utility creates an Essbase export model and assigns properties for each export model element. Various categories of properties are applicable at different levels:

● Essbase export Model properties
● Hierarchy properties
● Chain properties
● Level properties

Mapping Aliases, User-defined Members, and Reports

In the final stages of the migration process, the migration utility maps the following elements:

● Alias table sets are created.
● Aliases are defined according to their dimension elements.
● User-defined members in the Integration Services metaoutline are created as dimension elements in the Essbase Studio metaoutline folder.
● Drill-through reports in the Integration Services metaoutline are migrated along with intersection information, column lists, row governors, time governors, template SQLs, and other parameters. The drill-through reports are then associated with the export model.

Accessing the EIS Catalog Migration Dialog Box

➤ To access the EIS Catalog Migration dialog box:

1 Access the Tools menu in the Essbase Studio Console.
2 Left-click EIS Catalog Migration.

Migrating EIS Models and Metaoutlines

The Integration Services catalog you want to migrate to Essbase Studio does not have to reside on the same machine as Essbase Studio. The Integration Services catalog can be anywhere in an Integration Services-supported database on your network.
Even if the Integration Services catalog resides on the same machine as Essbase Studio, you must nevertheless create an ODBC DSN to the Integration Services catalog. This DSN must point to the database where the Integration Services catalog resides.

**Note:**
These procedures are addressed to database administrators and those who have a thorough knowledge of relational database concepts, Integration Services, and the relational databases in their working environment.

To migrate Integration Services models and metaoutlines from Integration Services to Essbase Studio:

1. If you have not already done so, access the EIS Catalog Migration dialog box. See “Accessing the EIS Catalog Migration Dialog Box” on page 255.
2. In the EIS Catalog DSN text box, enter the ODBC DSN which points to the Integration Services catalog you want to migrate.

   **Note:**
   If the ODBC DSN to the Integration Services catalog does not exist, you must create the ODBC DSN. If the Integration Services catalog does not reside on the same machine as Essbase Studio, you must create an ODBC DSN on the machine where the Essbase Studio client is running. This DSN must point to the database where the Integration Services catalog resides.
3. In the User text box, enter the user name for the EIS catalog; for example, tbc.
4. Enter your password for the EIS catalog.
5. Click the Fetch Models button.
   Essbase Studio accesses the Integration Services catalog and lists the models and metaoutlines in the catalog. Metaoutlines are grouped under their respective models.
6. In the Models and Data Sets column, select the Integration Services models and metaoutlines you want to migrate.

   **Note:**
   You can migrate multiple models and metaoutlines.
7. Click in the Data Source Connection column and, from the drop-down list, select the target relational database.

   **Note:**
   The target relational database should match the Integration Services target data source.

   **Optional:** You can create a target data source by clicking New Data Source and entering the name of the data source.
Note:
If you are migrating multiple models, you must select a target relational database in Essbase Studio for each model you are migrating.

8 Click in the Catalog column and, from the drop-down list, select an Integration Services catalog.

Note:
If your relational database is Oracle, selecting a catalog is not required.

9 Click in the Schema column and, from the drop-down list, select a schema.

Note:
If you are migrating multiple models, you must select a schema for each model you are migrating.

10 In the Select Folder text box, enter the target folder in Essbase Studio.

Optional: Click the Browse button to quickly find the target folder.

Optional: You can create a folder by entering the name of the folder in the Select Folder text box.

Note:
The target folders are also listed in the Metadata Navigator in Essbase Studio.

11 Click the Migrate button.

Your models and metaoutlines are migrated from the Integration Services catalog to the Essbase Studio catalog. When the migration is complete, the model is displayed in the Metadata Navigator. If you created a data source, the name of the data source is displayed in the Source Navigator.

Limitations and Restrictions
The Essbase Studio migration utility has the following limitations and restrictions.

Guidelines for Metaoutlines and Models
- Hybrid analysis-enabled metaoutlines are not migrated.
- Unicode-enabled metaoutlines are not migrated.
- All tables listed in your metaoutline must be present in the source database.
- The data types in the Integration Services model must match the data types in the source database. Essbase Studio verifies the data types, and if there is a mismatch, the migration will fail.
- Metaoutlines that have drill-through defined on recursive hierarchies can be migrated, but recursive-hierarchy drill-through functionality is not supported in Essbase Studio.
Member name columns for recursive hierarchies are not migrated.

- Metaoutlines with multiple ODBC DSNs are not migrated.
- Metaoutlines that have drill-through defined to alternate data sources can be migrated, but alternate-data sources drill-through functionality is not supported in Essbase Studio.
- Essbase Studio uses the bindings of dimension elements to create unique or duplicate outlines. By default, however, the migration studio migrates all metaoutlines as unique. To create duplicate outlines in Essbase Studio, you must modify the key bindings in the dimension elements by providing a key column. The data type of the key column must match the data type of the dimension element.

## Properties Not Migrated

In addition to the preceding items, there are several Integration Services properties that the Essbase Studio migration utility will not migrate. These properties include, but are not limited to, the following:

- Extra joins
- User-defined dataload SQL
- User-defined attributes

Note:

This limitation applies to transformation rules supplied through operators and pass-through SQL. However, some date-related transformations are supported: Q, DD, DM, DW, WM, WY, MMM, MM, MONTH, YY, and YYYY.

- Governors
- View sample-stop indicators
- Unique key columns
- Member name columns for recursive hierarchies
- Metaoutline level filters
- Overwriting

Note:

The above list is not inclusive; rather, it shows the most commonly used Oracle Essbase Integration Services properties which are not supported by the Essbase Studio migration utility.
Essbase Studio Accessibility Overview

Essbase Studio, though not fully accessible, does have some accessibility features built into it. These features focus on keyboard equivalents to some mouse actions. These features are always available, there is nothing to enable or disable in Essbase Studio in order for the keyboard equivalents to work.

See the following section for information on the various keyboard equivalents available in Essbase Studio.

Keyboard Equivalents

The following tables describe keyboard equivalents available in Essbase Studio:

- Table 6, “Accessing Elements in the Metadata Navigator and Source Navigator,” on page 260
- Table 7, “Accessing Editors, Wizards, and Dialog Boxes to Create New Elements,” on page 261
- Table 8, “Accessing Other Dialog Boxes,” on page 261
- Table 9, “Accessing the Context Menu in a Minischema Diagram,” on page 262
- Table 10, “Accessing Additional Minischema Context Menu Items When a Minischema Table is Selected,” on page 262
- Table 11, “Accessing Help Menu Items,” on page 262

<table>
<thead>
<tr>
<th>Table 5</th>
<th>Accessing Main Areas of the Essbase Studio Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
<td>Key or Key Combination</td>
</tr>
<tr>
<td>Access to the Essbase Studio menu bar.</td>
<td>F10</td>
</tr>
<tr>
<td>Focus on Metadata Navigator.</td>
<td>Ctrl+5</td>
</tr>
<tr>
<td>Action</td>
<td>Key or Key Combination</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>Focus on Minischema tab in Source Navigator.</td>
<td>Ctrl+4</td>
</tr>
<tr>
<td>Focus on Data Sources tab in Source Navigator.</td>
<td>Ctrl+3</td>
</tr>
<tr>
<td>Focus on Console Message tab.</td>
<td>Ctrl+2</td>
</tr>
<tr>
<td>Display the context menu appropriate to the focus of the cursor.</td>
<td>Shift+F10</td>
</tr>
<tr>
<td>Close, without saving, an open dialog box or wizard.</td>
<td>Esc</td>
</tr>
</tbody>
</table>

<p>| Table 6  Accessing Elements in the Metadata Navigator and Source Navigator |
|-----------------------------------------------|--------------------------------------------------|
| <strong>Action</strong>                                   | <strong>Key or Key Combination</strong>                        |
| Navigate through tree structure; right arrow to open a folder, left arrow to close a folder. | Use mouse to select item in tree, then use arrow keys |
| Context menu for tree items.                 | Shift+F10                                         |
| Launch the wizard, editor, or properties dialog box for the selected element. | Use mouse to select item in tree, press Enter |
| Open the Search tab in the Metadata Navigator. | Alt+E, then S for Search Metadata Elements |
| Open the Find dialog box.                    | Alt+E, then F for Find                           |
| Launch the wizard, editor, or properties dialog box for the selected metadata element. | Ctrl+5, use mouse to select element, Shift+F10, then E for Edit |
| Launch the Properties dialog box for the selected source element. | Ctrl+4 or Ctrl+3, use mouse to select element, Shift+F10, then P for Properties |
| Launch sample data for the selected metadata element in the work area. | Ctrl+5, use mouse to select a dimension element, Shift+F10, then V for View Sample Data |
| Launch sample data for the selected source element in the work area. | Ctrl+4 or Ctrl+3, use mouse to select a table or column, Shift+F10, then V for View Sample Data |
| Launch the lineage editor in the context of the selected metadata element. | Ctrl+5, use mouse to select element in the Metadata Navigator, Shift+F10, then S, S for Show Lineage |
| Launch the graphical view of the Essbase model in the work area.     | Ctrl+5, use mouse to select an Essbase model in the Metadata Navigator, Shift+F10, then E for Edit |
| Launch the Essbase Model Properties dialog box.                      | Ctrl+5, use mouse to select an Essbase model in the Metadata Navigator, Shift+F10, then E, E for Essbase Properties |
| Launch the Cube Deployment Wizard.                                  | Ctrl+5, use mouse to select an Essbase model in the Metadata Navigator, Shift+F10, then C for Cube Deployment Wizard |
| Launch the deployment history in the work area.                     | Ctrl+5, use mouse to select an Essbase model in the Metadata Navigator, Shift+F10, then S for Show Deployment History |</p>
<table>
<thead>
<tr>
<th>Action</th>
<th>Key or Key Combination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Launch the Incremental Update dialog box.</td>
<td>Ctrl+3, use mouse to select a data source connection, Shift+F10, then I for Incremental Update</td>
</tr>
<tr>
<td>Launch the Introspection Wizard.</td>
<td>Ctrl+3, use mouse to select a data source connection, Shift+F10, then I, I for Introspect</td>
</tr>
</tbody>
</table>

Table 7  Accessing Editors, Wizards, and Dialog Boxes to Create New Elements

<table>
<thead>
<tr>
<th>Action</th>
<th>Key or Key Combination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Launch the User-Defined Table dialog box.</td>
<td>Alt+F, followed by N, then U</td>
</tr>
<tr>
<td>Launch the Folder Properties dialog box.</td>
<td>Ctrl+5, mouse to appropriate location in the tree, Alt+f, followed by N, then F</td>
</tr>
<tr>
<td>Launch the Dimension Element dialog box.</td>
<td>Ctrl+5, mouse to appropriate location in the tree, Alt+f, followed by N, then D, D</td>
</tr>
<tr>
<td>Launch the Derived Text Measure dialog box.</td>
<td>Ctrl+5, mouse to appropriate location in the tree, Alt+f, followed by N, then D, D, D</td>
</tr>
<tr>
<td>Launch the hierarchy editor.</td>
<td>Ctrl+5, mouse to appropriate location in the tree, Alt+f, followed by N, then H</td>
</tr>
<tr>
<td>Launch the measure hierarchy editor.</td>
<td>Ctrl+5, mouse to appropriate location in the tree, Alt+f, followed by N, then M, M</td>
</tr>
<tr>
<td>Launch the Calendar Hierarchy dialog box.</td>
<td>Ctrl+5, mouse to appropriate location in the tree, Alt+f, followed by N, then C</td>
</tr>
<tr>
<td>Launch the Cube Schema Wizard.</td>
<td>Ctrl+5, mouse to appropriate location in the tree, Alt+f, followed by N, then C, C</td>
</tr>
<tr>
<td>Launch the drill-through report editor.</td>
<td>Ctrl+5, mouse to appropriate location in the tree, Alt+f, followed by N, then D, D, D</td>
</tr>
<tr>
<td>Launch the Create Date Elements dialog box.</td>
<td>Ctrl+5, mouse to appropriate location in the tree to select a date element, Shift+F10, then c, c for Create Date Elements</td>
</tr>
</tbody>
</table>

Table 8  Accessing Other Dialog Boxes

<table>
<thead>
<tr>
<th>Action</th>
<th>Key or Key Combination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Launch the Preferences dialog box.</td>
<td>Alt+E, then P</td>
</tr>
<tr>
<td>Launch the Alias Set Manager.</td>
<td>Alt+T, then A</td>
</tr>
<tr>
<td>Launch the Update Cube Linkage dialog box.</td>
<td>Alt+T then U</td>
</tr>
<tr>
<td>Launch the EIS Catalog Migration dialog box.</td>
<td>Alt+T, then E</td>
</tr>
</tbody>
</table>
Table 9   Accessing the Context Menu in a Minischema Diagram

<table>
<thead>
<tr>
<th>Action</th>
<th>Key or Key Combination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activate the context menu.</td>
<td>Shift+F10, down arrow key once</td>
</tr>
<tr>
<td>Save the active minischema.</td>
<td>Shift+F10, down arrow key once, Enter</td>
</tr>
<tr>
<td>Launch the Add Join dialog box.</td>
<td>Shift+F10, down arrow key once, then A</td>
</tr>
<tr>
<td>Launch the Add Joins by Inspection dialog box.</td>
<td>Shift+F10, down arrow key once, then A, A</td>
</tr>
<tr>
<td>Launch the Edit Properties dialog box (this is the Add/Remove Table command on the context menu)</td>
<td>Shift+F10, arrow key once, then A, A, A</td>
</tr>
<tr>
<td>Lay out the minischema in the work area.</td>
<td>Shift+F10, then L</td>
</tr>
<tr>
<td>Zoom in on, or enlarge the view of, the minischema.</td>
<td>Ctrl+=</td>
</tr>
<tr>
<td>Zoom out on, or reduce the view of, the minischema</td>
<td>Ctrl+-</td>
</tr>
<tr>
<td>Launch the Properties dialog box for the minischema</td>
<td>Shift+F10, then P</td>
</tr>
</tbody>
</table>

Table 10   Accessing Additional Minischema Context Menu Items When a Minischema Table is Selected

<table>
<thead>
<tr>
<th>Action</th>
<th>Key or Key Combination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Move focus from one table to another in the minischema diagram</td>
<td>Arrow keys</td>
</tr>
<tr>
<td>Launch the Choose Folder dialog. Use the arrow keys to navigate to the folder where you want to add the metadata element created from the selected minischema table.</td>
<td>Shift+F10, then A A A A</td>
</tr>
<tr>
<td>Display a drop-down list of colors, use arrow keys then press Enter to select a color</td>
<td>Shift+F10, then C</td>
</tr>
<tr>
<td>Launch sample data in the work area for the selected table</td>
<td>Shift+F10, then V</td>
</tr>
<tr>
<td>Launch the Properties dialog box for the selected table</td>
<td>Shift+F10, then P</td>
</tr>
</tbody>
</table>

Table 11   Accessing Help Menu Items

<table>
<thead>
<tr>
<th>Action</th>
<th>Key or Key Combination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Launch the Welcome page in the work area</td>
<td>Alt+H, then W</td>
</tr>
<tr>
<td>Launch the Essbase Studio online help</td>
<td>Alt+H, then H</td>
</tr>
<tr>
<td>Launch the Essbase Studio information map</td>
<td>Alt+H, then I</td>
</tr>
<tr>
<td>Launch the Oracle Support website</td>
<td>Alt+H, then O</td>
</tr>
<tr>
<td>Launch the Oracle Technology Network website</td>
<td>Alt+H, then O, O</td>
</tr>
<tr>
<td>Launch the Oracle website</td>
<td>Alt+H, then O, O, O</td>
</tr>
<tr>
<td>Launch the Essbase Studio About box</td>
<td>Alt+H, then A</td>
</tr>
</tbody>
</table>
In This Appendix

Overview ....................................................................................................................... ...263
Guidelines for Accessing Essbase Studio Server.................................................................263
Catalogs and Data Sources Guidelines ...............................................................................263
Introspection Limitations ..................................................................................................265
Metadata Elements Usage Limitations ...................................................................................265
Hierarchies Usage Limitations .............................................................................................266
Cube Schemas Limitations ..................................................................................................266
Essbase Properties Editing and Usage Limitations ..............................................................267
Cube Deployment Limitations ...............................................................................................267
Drill-through Reports Limitations ........................................................................................268

Overview

This appendix describes various limitations you may encounter while working with Essbase Studio.

Guidelines for Accessing Essbase Studio Server

Access issues can arise when multiple clients are making requests to an instance of Essbase Studio Server if one of the clients is in the process of scraping a data source for metadata elements. When scraping a data source, Essbase Studio requires exclusive access to the catalog in order to add the metadata derived from scraping. Requests from other Essbase Studio clients—such as login requests, viewing or editing hierarchies or dimension elements—must wait until scraping is completed in order for Essbase Studio Server to process other requests.

You can configure the amount of time clients must wait before being prompted that the request cannot be executed. See “server.readLockTimeOut” on page 32 and “server.writeLockTimeOut” on page 33 for more information.

Catalogs and Data Sources Guidelines

The following topics describe limitations involving the Essbase Studio catalog, physical data sources, and data source connections.
Catalog and Data Source Permission Guidelines

- The Essbase Studio catalog database user must have at least write permission to the Essbase Studio catalog database. The catalog database user is the user specified in the `catalog.username` and `catalog.password` properties in the `server.properties` file. This user must have at least write permission to the database specified in the `catalog.db` property.

- If Oracle is your catalog database: Each time you start Essbase Studio Server, the server will try to execute the following ALTER SYSTEM statement:

  ```sql
  ALTER SYSTEM SET open_cursors=300 SCOPE=MEMORY
  ```

  If the catalog user (specified in the `server.properties` file during configuration) does not have the privileges necessary to execute the ALTER SYSTEM statement, Essbase Studio will log a warning message and continue working.

- When setting up a data source connection in the Connection Wizard, Essbase Studio users must specify a user name and password that has at least read permission to the external data source (database) to which they are connecting.

- Oracle recommends that users have only read permission to external data sources in order to prevent problems resulting from the use of Template SQL for drill-through reports and custom data load SQL.

Oracle Client Driver Guidelines

The DataDirect Oracle ODBC driver uses a connection string format that differs from the Microsoft Oracle ODBC driver. The connection string format used for Essbase Studio is designed for the DataDirect Oracle ODBC driver.

Note the following guidelines for using non-DSN connections:

- The non-DSN connection to Oracle is intended for use with the DataDirect Wire Protocol ODBC driver only.

- The non-DSN connection to IBM DB2 is intended for use with the DataDirect Wire Protocol ODBC driver only.

- The non-DSN connection to Microsoft SQL Server is intended for use with both the Microsoft SQL Server ODBC driver and the DataDirect Wire Protocol ODBC driver.

- The non-DSN connection to Teradata is intended for use with the native Teradata ODBC driver only. It is not intended for use with the DataDirect Wire Protocol ODBC driver.
MySQL Version 5.0.17 Patched to 5.0.22
Oracle advises that you do not use MySQL server version 5.0.17 patched to 5.0.22. Problems with queries containing the DISTINCT clause in this patched version cause MySQL server to cut off query results during conversion from integer to string.

Excel Files as Data Source Not Supported
Essbase Studio does not support Microsoft Excel files as data sources.

General Catalog and Data Source Limitations and Guidelines
● Text file data source directory names must not contain spaces.
● Single quotes (‘) are not allowed in these data source entities:
  ○ Relational data source schema, catalog, table, or column names.
  ○ Text file data source directory, file, or column names.
● Non-English characters are not supported in DSN names.

Introspection Limitations
● Introspection is not supported for these data sources:
  ○ Text files
  ○ Oracle Business Intelligence

Metadata Elements Usage Limitations
The following topics describe limitations in the usage of metadata elements.
● “Derived Text Measures Limitations” on page 265
● “Cycle Dependency Guidelines” on page 265

Derived Text Measures Limitations
● You cannot create derived text measures from elements in text file connections.
● When derived text measures are used in cube schemas to build Essbase models, the “XOLAP Model” option will not be available for the model.

Cycle Dependency Guidelines
Essbase Studio does not verify cycle dependencies between metadata elements. Cycle dependency is a relationship between metadata elements that cycles back upon itself.
For example, suppose Metadata Element A uses Metadata Element B in its expression definition; Metadata Element B uses Metadata Element C in its expression definition; and Metadata Element C uses Metadata Element A in its expression definition. The expressions defining these elements have a cycle dependency relationship, and this is not verified by Essbase Studio.

If you have created metadata elements with cycle dependency, you are responsible for maintaining the validity of the cycle dependency between those elements.

**Hierarchies Usage Limitations**

The following topics describe limitations in using hierarchies:

- “Calendar Hierarchies Limitations” on page 266

**Standard and Measure Hierarchies Limitations**

These limitations apply to standard and measure hierarchies:

- You cannot add a data source column or metadata element into a hierarchy chain if the column or element already exists in the chain.

- Use care when dragging and dropping multiple data source columns or metadata elements into a hierarchy chain. If one or more columns or elements are already present in the chain, the drop is rejected. When dragging multiple columns or elements into a hierarchy chain, be sure to select columns or elements that are not already present in the chain.

**Calendar Hierarchies Limitations**

These limitations apply to calendar hierarchies:

- When an Essbase model contains a dimension created from a Calendar hierarchy, these limitations apply:
  - Text file data sources are not supported.
  - During cube deployment, in the “Load data options” group, selecting the “Overwrite existing data” option is not allowed.

**Cube Schemas Limitations**

These limitations apply to cube schemas:

- If you add data load bindings to an existing cube schema from which an Essbase model has already been created, the cube schema and model will be out of sync. If you deploy from the existing Essbase model, the model will not pick up the new data load bindings, resulting in invalid deployment results. To deploy a cube using the new data load bindings, you must create a new Essbase model and deploy from the new model.
Essbase Properties Editing and Usage Limitations

The following topics describe limitations in the editing and usage of Essbase model properties:

- “Custom Data Load SQL Guidelines” on page 267
- “Independent Dimension Bindings Limitations” on page 267
- “XOLAP Functionality Guidelines” on page 267

Custom Data Load SQL Guidelines

Oracle does not recommend using custom data load SQL for Essbase models that support duplicate member names. Essbase Studio generates special tags internally to uniquely identify the duplicate members.

Independent Dimension Bindings Limitations

These limitations apply to defining the bindings of independent dimensions for use in varying attributes:

- The expression to define the binding for an independent dimension can be applied to either unique or duplicate outline deployments, but the expression should not be applied to both unique and duplicate members at the same time.
- For a unique outline deployment, the result of the expression must match the caption binding of the class of the independent members.
- For a duplicate outline deployment, the result of the expression must match the key binding of the class of the independent members.

XOLAP Functionality Guidelines

These guidelines apply to XOLAP functionality:

- Alternate hierarchies are supported by XOLAP.
- A dimension with alternate hierarchies is tagged as “Multiple Hierarchy Enabled” automatically.
- Attribute dimensions are supported.
- Attribute hierarchies are not supported. An attribute dimension can have only one child level.

Cube Deployment Limitations

These limitations apply to cube deployment:

- When loading data from an Essbase model containing a dimension created from a Calendar hierarchy, these limitations apply:
Text file data sources are not supported.

In the “Load data options” group, selecting the “Overwrite existing data” option is not allowed.

- If you add data load bindings to an existing cube schema from which an Essbase model has already been created, the cube schema and model will be out of sync. If you deploy from the existing Essbase model, the model will not pick up the new data load bindings, resulting in invalid deployment results. To deploy a cube using the new data load bindings, you must create a new Essbase model and deploy from the new model.

- If you are deploying from an Essbase model that is enabled for XOLAP, you must specify a new application and database name each time you deploy. You cannot deploy over an existing application and database.

- In nonstreaming mode (server.essbase.streamingCubeBuilding=false in server.properties), Essbase Studio can only deploy cubes from a version 10.1.3.4 or above of Oracle Business Intelligence (OBI) source. Cubes may be deployed from an earlier version of OBI, 10.1.3.3, only if the server.essbase.streamingCubeBuilding property is set to streaming (server.essbase.streamingCubeBuilding=true). See “server.essbase.streamingCubeBuilding” on page 28 for information on this property.

**Drill-through Reports Limitations**

The following topics describe limitations involving drill-through reports:

- “Guidelines for Accessing Drill-through Reports in Spreadsheet Add-in” on page 268
- “General Drill-through Operations Limitations” on page 269

**Guidelines for Accessing Drill-through Reports in Spreadsheet Add-in**

- In Spreadsheet Add-in, to access drill-through reports from cubes built through Essbase Studio, users must be registered with Essbase Studio and be provisioned with, at a minimum, the Essbase Read role and the Essbase Studio cpViewer role.

- When logging in to Essbase in Spreadsheet Add-in, login names and password are case-sensitive for users who are provisioned through Oracle’s Hyperion® Shared Services.

- Essbase Studio Server should be running in order to access drill-through reports on cubes created using Essbase Studio. If Essbase Studio Server is not running:
  - In Spreadsheet Add-in, a message is returned stating that there are no drill-through reports present.
  - In Oracle Hyperion Smart View for Office, Fusion Edition, a message is returned stating that Essbase Studio Server is not running.

- In Oracle Essbase Spreadsheet Add-in, cubes built by Essbase Studio do not display the visual clues that denote drill-through cells. Because there is no way to know for certain which particular cells are drill-through cells, any data cell in the spreadsheet is a potential drill-
through cell. To discover if a drill-through report is available on a particular cell, perform these steps:

1. Select a data cell, and then access the Linked Object Browser (either by selecting Linked Object Browser from the Essbase menu, or by double-clicking the cell).

   Once launched, the browser automatically displays an entry indicating that a drill-through report is available from Integration Server. This is true whether or not a drill-through report is actually available from the selected cell.

2. Select the Drill-Through object in the list in the Linked Object Browser and then click View/Launch.

   If one or more drill-through reports are available for the selected cell, the Available Reports dialog box is displayed listing those reports. In Available Reports, select the drill-through report to view and click Execute.

   If no drill-through reports are available for the selected cell, a message is displayed indicating that there are no reports are defined for the specified intersection.

---

**General Drill-through Operations Limitations**

- These drill-through operations are not supported:
  - Drill-through from dimensions built using recursive hierarchies.
  - Drill-through from multi-cell range.

- Oracle does not recommend associating drill-through reports that use drill-through Template SQL with Essbase models that support duplicate member names. Essbase Studio generates special tags internally to uniquely identify the duplicate members.

- Drill-through target URLs cannot exceed 256 characters in length.

- Hierarchies created from Oracle Hyperion EPM Architect, Fusion Edition sources are not supported for drill-through reports.
Naming Restrictions for Databases and Applications

The following Essbase naming conventions apply to database and application names:

- 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.
- Do not use the characters listed in Table 12.

Table 12  List of Restricted Characters in Application and Database Names

<table>
<thead>
<tr>
<th>Character</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>asterisk</td>
</tr>
<tr>
<td>[ ]</td>
<td>brackets</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>equal sign</td>
</tr>
<tr>
<td>&gt;</td>
<td>greater-than sign</td>
</tr>
<tr>
<td>&lt;</td>
<td>less-than sign</td>
</tr>
<tr>
<td>.</td>
<td>period</td>
</tr>
<tr>
<td>+</td>
<td>plus sign</td>
</tr>
<tr>
<td>?</td>
<td>question mark</td>
</tr>
</tbody>
</table>

In This Appendix

- Naming Restrictions for Databases and Applications 271
- Naming Restrictions for Metadata Elements 272
- Drill-through Reports Naming Guidelines 276
- Exceptions to Essbase Naming Restrictions 276
<table>
<thead>
<tr>
<th>Character</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>double quotation mark</td>
</tr>
<tr>
<td>'</td>
<td>single quotation mark</td>
</tr>
<tr>
<td>/</td>
<td>forward slash</td>
</tr>
<tr>
<td>\</td>
<td>backslash</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- For database names, do not use the:
  - String drxxxxxx (not case-sensitive)
  - Reserved word Replay
- For aggregate storage databases, do not use the following words as application or database names:
  - DEFAULT
  - LOG
  - METADATA
  - REPLAY
  - TEMP

Application and database names are case-sensitive. 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 Metadata Elements**

The following topics discuss the Essbase naming conventions as they apply to metadata elements:

- “Metadata Elements Subject to Naming Conventions” on page 272
- “Metadata Element Naming Restrictions” on page 273

**Metadata Elements Subject to Naming Conventions**

Essbase naming conventions apply to the following metadata elements:

- Dimension names
- Member names
- Aliases
- Derived text measures
- Any dimension element that is not column based; that is, binding in a string constant
- Any dimension element used in a measures hierarchy
Metadata Element Naming Restrictions

The following topics list naming restrictions as they apply to metadata elements:

- “General Metadata Element Naming Guidelines” on page 273
- “Restricted Characters” on page 273
- “Reserved Words” on page 274

General Metadata Element Naming Guidelines

When naming metadata elements, follow these guidelines:

Note:

When in doubt about a metadata element name in Essbase Studio, it is generally safe to follow the Essbase naming conventions; however, see “Exceptions to Essbase Naming Restrictions” on page 276.

- Use no more than 80 bytes when naming metadata elements.
- Names are not case-sensitive unless case-sensitivity is enabled.
  
  See “Setting Outline Properties” in the Oracle Essbase Administration Services Online Help.
- Even when case-sensitivity is enabled in an aggregate storage outline for which duplicate member names is enabled, do not use the same name with only case differences for a metadata element name. For example, do not name two dimensions “Product” and “product.”
- Do not use quotation marks (" "), brackets ([]), or tabs in a name.
- Do not place spaces at the beginning or end of a name. Essbase ignores such spaces.
- Calculation script commands, operators, and keywords
- Report writer commands
- Function names and function arguments
- Names of other metadata elements (unless a member is shared)

Note:

If you enable Dynamic Time Series members, do not use the associated generation names—History, Year, Season, Period, Quarter, Month, Week, or Day.

Restricted Characters

At the beginning of a dimension or member name, do not use the characters listed in Table 13:
Table 13  List of Restricted Characters for Metadata Element Names

<table>
<thead>
<tr>
<th>Character</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>@</td>
<td>at sign</td>
</tr>
<tr>
<td>\</td>
<td>backslash</td>
</tr>
<tr>
<td>{}</td>
<td>brace</td>
</tr>
<tr>
<td>,</td>
<td>comma</td>
</tr>
<tr>
<td>-</td>
<td>dash, hyphen, or minus</td>
</tr>
<tr>
<td>=</td>
<td>equal sign</td>
</tr>
<tr>
<td>&lt;</td>
<td>less than sign</td>
</tr>
<tr>
<td>()</td>
<td>parentheses</td>
</tr>
<tr>
<td>.</td>
<td>period</td>
</tr>
<tr>
<td>+</td>
<td>plus sign</td>
</tr>
<tr>
<td>'</td>
<td>single quotation mark</td>
</tr>
<tr>
<td>_</td>
<td>underscore</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Reserved Words**

Do not use any of the following words in the names of metadata elements:

ALL
AND
ASSIGN
AVERAGE
CALC
CALCMBR
COPYFORWARD
CROSSDIM
CURMBRNAME
DIM
DIMNAME
DIV
DYNAMIC
EMPTYPARAM
EQ
EQOP
EXCEPT
EXP
EXPERERROR
FLOAT
FUNCTION
GE
GEN
GENRANGE
GROUP
GT
ID
IDERROR
INTEGER
LE
LEVELRANGE
LOOPBLOCK
LOOPPARMS
LT
MBR
MBRNAME
MBRONLY
MINUS
MISSING
MUL
MULOP
NE
NON
NONINPUT
NOT
OR
PAREN
PARENPARM
PERCENT
PLUS
RELOP
SET
SKIPBOTH
SKIPMISSING
SKIPNONE
SKIPZERO
TO
TOLOCALRATE
TRAILMISSING
TRAILSUM
UMINUS
UPPER
VARORXMBR
XMBRONLY
$$\text{UNIVERSE}$$
Drill-through Reports Naming Guidelines

- When naming drill-through reports, follow these guidelines:
  - Cannot end with a backslash (\)
  - Cannot contain the pipe character ( | )
  - Cannot contain a single quote (')

Exceptions to Essbase Naming Restrictions

Most Oracle Essbase naming restrictions apply to Essbase Studio; however, under certain circumstances, some Essbase Studio entities have naming conventions which differ:

- Folders
- Data sources
- Minischemas
- Cube schemas
CPL Expressions Overview

Some tools in Oracle Essbase Studio, such as the Drill-through Filter and Edit Properties for dimension elements, enable you to write CPL (Common Platform Language) expressions for accessing information from data sources. An expression is a sequence of operands and operators following the language-defined syntax. Each expression returns a value, the type of which defines the type of the expression.

Note:
All subexpressions in a CPL expression must come from the same connection.

CPL expressions are composed of:
- “Operands” on page 277
- “Operators” on page 281

Operands

Operands are the objects on which expressions operate. They are names used for representing and accessing data.

Syntax Elements

Operands can contain the following elements, with syntax varying according to the value type and data source:
- <KEYWORD>
A keyword indicating a value type. For example, a keyword would indicate if the value type of the expression is a connection or a class.

- :  
  The colon (".") assigns the value type in the keyword to the expression.

- \  
  The backslash ("\") indicates a folder. Folders are objects combining other objects, including other folders, defining the location of those objects in a catalog. A folder cannot contain two objects of the same name and same type, but can contain two objects of the same name but different types.

  The folder can be expressed by:
  ```
  <folder> ::= \ | <folder> <folder_name> \
  ```

  Accessing the root folder:
  ```
  \ 
  ```

  Accessing a folder:
  ```
  \ 'folder'\
  ```

  Accessing a subfolder:
  ```
  \ 'folder'\'subfolder'\
  ```

- 'NAME'  
  Element names must be in single quotes. Depending on the data source, names can be connections, classes, catalogs, schema, tables, folders, or subfolders.

- ::
  The double-colon ("::") accesses database connections or text files. For example, to access information in a database connection called 'tbc':
  ```
  connection : \ 'tbc':.'tbc.family'.'FAMILY'
  ```

- .
  The dot (".") accesses attributes or values contained in tables or files, such as columns.

**Connections**

Connections access physical data sources to retrieve data, such as relational tables, multidimensional cubes, or flat files. The connection keyword indicates a connection, and follows a different syntax depending on the source.

**Note:**

Connections can exist only in the root folder.
Databases with Two Levels
A database connection may have two levels:

connection : '<CONNECTION_NAME>':'<CATALOG/SCHEMA>.<TABLE>':'<COLUMN>'

For example:
connection : 'tbc':'tbc.family':'FAMILY'

Databases with Three Levels
A database connection may have three levels:

connection : '<CONNECTION_NAME>':'<CATALOG>.<SCHEMA>.<TABLE>':'<COLUMN>'

For example:
connection : 'tbc':'tbc.tbc.family':'FAMILY'

Flat File (One-Level)
A flat file has only one level:

connection : '<DATA_SOURCE_NAME>':'<FILENAME>':'<COLUMN>'

For example:
connection : 'DataFile':'dmdemo.txt':'Actual'

User-Defined Table
A user-defined table can have any name, but the name cannot contain single quotes:

connection : '<DATA_SOURCE_NAME>':'<USER_DEFINED_FILE_NAME>':'<COLUMN>'

For example:
connection : 'tbc':'myTable':'FAMILY'

Classes
A class is a logical element expressing a business entity, and you can bind them to physical data sources. Simple classes can become the basis for more complex classes. Logical elements have a unit, such as: integer, number, or string. Each class has a binding to an external data source to express that data.

The class keyword indicates a class, and follows this general pattern:

class : <folder> '<CLASS_NAME>'

For example:
class : 'folder1' 'folder2''My Object'
Note:

Classes can exist in any folder.

**Constants**

Constants and constant literals are objects storing temporary values. Constants express basic value types, while constant literals describe values of the complex value types. Constants are referred to through hierarchies or classes.

- **Number constant**
  
The number constant has the unit number.
  
  Unsigned positive:
  
  31.27
  
  Signed negative:
  
  -10.4
  
  Scientific:
  
  5.4e+4

- **Integer constant**
  
The integer constant has the unit integer.
  
  Unsigned positive:
  
  1
  
  Signed negative:
  
  -45

- **Long constant**
  
The long constant has the unit long.
  
  Example:
  
  1L
  
  Example:
  
  100000000000001L

- **Boolean constant**
  
The boolean constant has the unit boolean.
  
  A boolean returns either true or false depending on the evaluated condition.

- **String constant**
  
The string constant has the unit string. There are two ways to express string constants:
  
  - With quotes:
    
    "Example of string value"
If you want double quotes inside a string, you must double the double-quote character within the string, for example:

"Example of string value "" with quote inside"

- With square brackets:
  $\{\text{Example of string value } " \text{ with quote inside}\}$

- With parentheses:
  $\{\text{(Example of string value } " \text{ with quote inside)}\}$

String constants in square brackets and parentheses can contain any symbols in the string, such as quotes, and they will behave as characters, without being interpreted by the parser for special use.

**Operators**

Operators are the commands expressions perform on operands that return values. Each operator has a name, priority, and a signature. The signature defines the type values of a number of operands and their value types.

**Order of Operations**

Multiple operators are evaluated in the order shown in Table 14. Priority 1 is the highest priority; priority 8 is the lowest.

<table>
<thead>
<tr>
<th>Priority</th>
<th>Operators</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( )</td>
</tr>
<tr>
<td>2</td>
<td>not, +, – (Unary)</td>
</tr>
<tr>
<td>3</td>
<td>*, /, mod</td>
</tr>
<tr>
<td>4</td>
<td>+, – (Binary)</td>
</tr>
<tr>
<td>5</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>&lt;, &lt;=, &gt;, &gt;=, ==, !=</td>
</tr>
<tr>
<td>7</td>
<td>and</td>
</tr>
<tr>
<td>8</td>
<td>or</td>
</tr>
</tbody>
</table>

Operands with the same priority are executed from left to right.

**Grouping**

You use grouping operators to associate subexpressions.
Grouping—parentheses

Overrides the standard operator level of precedence. Can also be used to clarify an element of a formula, for example, around an operand to aid readability.

Example:

\((7-2)*6\)

Examples

Surrounding an Operand to Aid Readability

\(\text{connection : 'DB_conn'::'TBC.tbc.SALESFACT'.'OPENINGINVENTORY'}\)

With Connections

\((\text{connection : 'DB_conn'::'TBC.tbc.SALESFACT'.'OPENINGINVENTORY'}}\) - \(\text{connection : 'DB_conn'::'TBC.tbc.SALESFACT'.'SALES'}}\) \* \(\text{connection : 'DB_conn'::'TBC.tbc.SALESFACT'.'COGS'}}\)

Logical Operators

You use logical operators to evaluate conditions.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>not</td>
<td>Returns:</td>
<td>True—when a condition is false</td>
<td>True—if both conditions are true</td>
<td>True—if both conditions are true</td>
</tr>
<tr>
<td></td>
<td>● False—when a condition is true</td>
<td>● False—if either condition is false</td>
<td>● False—if either condition is false</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>not true</td>
<td>true and false</td>
<td>or</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Returns:</td>
</tr>
</tbody>
</table>
- True—if either condition true
- False—if both conditions are false

Example:
true or false

==
Condition—is equal to
Example:
2 == 1

!=
Condition—is not equal to
Example:
"Denver" != "Maine"

<
Ordinal condition—less than
Example:
0 < -0.1

>
Ordinal condition—more than
Example:
10 > 0.1

<=
Ordinal condition—less than or equal to
Example:
0 <= 50

>=
Ordinal condition—more than or equal to
Example:
10 >= 0

Examples

With Classes
class : \'folder1\'\'FAMILY' != "100"

With Connections
class : \'folder1\'\'DB_conn':::'TBC.tbc.SALESFACT'.SALES < 25000

Mathematical Operators
Mathematical operators perform arithmetic.
Addition or positive—commutative.
  ● **Unary**: positive integer
    Example:
    +1
  ● **Binary**: addition
    Example:
    2 + 3

Subtraction or negative—non-commutative.
  ● **Unary**: negative integer
    Example:
    -1
  ● **Binary**: subtraction
    Example:
    3 - 2

Multiplication—commutative.
Example:
5 * 10

Division—non-commutative.
Example:
100 / 10

Mod
Modulus (integer)—returns the remainder of two divisibles.
Example:
18 mod 5
returns 3.

Examples

With Connections

```cpl
(connection : \\
  \'DB_conn\':\'TBC.tbc.SALESFACT\'.\'SALES\')
```

With Grouping Operators

```cpl
(
  {connection :
    \'DB_conn\':\'TBC.tbc.SALESFACT\'.\'OPENINGINVENTORY\'}
```
String Operators

You use string operators on strings.

<table>
<thead>
<tr>
<th>Concatentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>It can only be applied to string expressions or sets. It combines multiple strings into one. It is not commutative.</td>
</tr>
</tbody>
</table>

Examples

Building Strings with Connections

connection : \'tbc\':\'tbc.region\'.\'UDA\'
  || \"\" ||
  || connection : \'tbc\':\'tbc.region\'.\'REGION\'

Building Strings with Converted Number Values

"Sales " ||
  (connection : \'tbc\':\'tbc.salesfact\'.\'SALES\'. toString)

CPL SQL Functions

SQL functions support standard SQL conventions.

Date

Datetime functions for SQL.

dayOfMonth

Returns the number of the day of the month of a given date.

Syntax

'dayOfMonth' ( <datetime> )

Parameters

- <datetime> is a datetime operand.
Returns

Integer

Example

'dayOfMonth' ( connection : \'DB'::'TBC.tbc.SALES'.'TRANSDATE' )

month

Returns the number of the month of the year in the given date.

Syntax

'month' ( <datetime> )

Parameters

● <datetime> is a datetime operand.

Returns

Integer

Example

'month' ( connection : \'DB'::'TBC.tbc.SALES'.'TRANSDATE' )

monthName

Returns the name of the month for a given date.

Syntax

'monthName' ( <datetime> )

Parameters

● <datetime> is a datetime operand.

Returns

String

Example

'monthName' ( connection : \'DB'::'TBC.tbc.SALES'.'TRANSDATE' )

monthShortName

Returns the abbreviated name of the month for a given date.
Syntax
'monthShortName' ( <datetime> )

Parameters
● <datetime> is a datetime operand.

Returns
String

Example
'monthShortName' ( connection : \'DB'::'TBC.tbc.SALES'.'TRANSDATE' )

**quarter**
Returns the number of the quarter of the year for a given date.

Syntax
'quarter' ( <datetime> )

Parameters
● <datetime> is a datetime operand.

Returns
Integer

Example
'quarter' ( connection : \'DB'::'TBC.tbc.SALES'.'TRANSDATE' )

**quarterAsString**
Returns the name of the quarter of the year for a given date.

Syntax
'quarterAsString' ( <datetime> )

Parameters
● <datetime> is a datetime operand.

Returns
String
Example

'quarterAsString' ( connection : 'DB'::'TBC.tbc.SALES'.'TRANSDATE' )

**weekday**

Returns the number of the day of the week for a given date.

Syntax

'weekday' ( <datetime> )

Parameters

- <datetime> is a datetime operand.

Returns

Integer

Example

'weekday' ( connection : 'DB'::'TBC.tbc.SALES'.'TRANSDATE' )

**weekdayName**

Returns the name of the day of the week for a given date.

Syntax

'weekdayName' ( <datetime> )

Parameters

- <datetime> is a datetime operand.

Returns

string

Example

'weekdayName' ( connection : 'DB'::'TBC.tbc.SALES'.'TRANSDATE' )

**weekdayShortName**

Returns the short name of the day of the week for a given date.

Syntax

'weekdayShortName' ( <datetime> )
Parameters
- `<datetime>` is a datetime operand.

Returns
- string

Example
```
'weekdayShortName' ( connection : '\DB'::'TBC.tbc.SALES'.'TRANSDATE' )
```

**year**
Returns the year of a given date.

Syntax
```
'year' ( <datetime> )
```

Parameters
- `<datetime>` is a datetime operand.

Returns
- string

Example
```
'year' ( connection : '\DB'::'TBC.tbc.SALES'.'TRANSDATE' )
```

**yearShort**
Returns the abbreviated year of a given date.

Syntax
```
'yearShort' ( <datetime> )
```

Parameters
- `<datetime>` is a datetime operand.

Returns
- string

Example
```
'yearShort' ( connection : '\DB'::'TBC.tbc.SALES'.'TRANSDATE' )
```
**String**

Strings follow SQL string conventions, so the first position in the string is 1. For example, the “A” is in position 1 in the following string, with each position noted under the example:

```
<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
</tbody>
</table>
```

**contains**

Determines if the substring exists in the full string.

**Syntax**

`contains ( <fullStr> , <substr> )`

**Parameters**

- `<fullStr>` is the full string.
- `<substr>` is a substring to find contained in the full string.

**Returns**

Boolean

- true—if the substring is in the string
- false—if not

**Example**

```
contains ( connection : '\DB'::'TBC.tbc.MARKET'.'STATE' , "e" )
```

**index**

Returns the starting number position of the substring within the full string.

**Syntax**

`index ( <fullStr> , <subStr> )`

**Parameters**

- `<fullStr>` is the full string.
- `<subStr>` is a substring to find contained in the full string.

**Returns**

Integer
Example

index ( connection : 'DB':'TBC.tbc.MARKET'.'STATE' , "e" )

**leftStr**

Returns a substring of the input string to the left of the index of the input number.

**Syntax**

leftStr ( <string> , <index> )

**Parameters**

- <string> is a string.
- <index> is an index number.

**Returns**

String

**Example**

leftStr ( connection : 'DB':'TBC.tbc.MARKET'.'STATE' , 2 )

**length**

Returns the number of characters, including spaces, in the input string.

**Syntax**

length ( <string> )

**Parameters**

- <string> is a string.

**Returns**

Integer

**Example**

length ( connection : 'DB':'TBC.tbc.MARKET'.'STATE' )

**lower**

Converts all alphabetic characters in the input string to lower case.

**Syntax**

lower ( <string> )
Parameters
- `<string>` is a string.

Returns
String

Example
`lower ( connection : \'DB'::'TBC.tbc.MARKET'.\'STATE' )`

**lTrim**

Removes leading spaces from the input string.

Syntax
`lTrim ( <string> )`

Parameters
- `<string>` is a string.

Returns
String

Example
`lTrim ( connection : \'DB'::'TBC.tbc.MARKET.STATE' )`

**rightStr**

Returns a substring of the input string to the right of the index of the input number.

Syntax
`rightStr ( <string> <index> )`

Parameters
- `<string>` is a string.
- `<index>` is an index number.

Returns
String

Example
`rightStr ( connection : \'DB'::'TBC.tbc.MARKET'.\'STATE' , 2 )`
**rTrim**

Removes trailing spaces from the input string.

**Syntax**

rTrim ( <string> )

**Parameters**

- <string> is a string.

**Returns**

String

**Example**

rTrim ( connection : 'DB':'TBC.tbc.MARKET'.STATE )

**soundex**

Returns a phonetic expression of the input string.

**Syntax**

soundex ( <string> )

**Parameters**

- <string> is a string.

**Returns**

String

**Example**

soundex ( connection : 'DB':'TBC.tbc.MARKET'.STATE )

**substr**

A substring of the input string.

**Syntax**

substr ( <string> , <startNumber> , <numberOfCharacters> )

**Parameters**

- <string> is a string.
- <startNumber> is a number representing the start position of the substring.
- <numberOfCharacters> is the length of the substring.
Returns
String

Example
substr ( connection : \'DB''::'TBC.tbc.MARKET'.'STATE' , 1 , 5 )

**trim**
Removes leading and trailing spaces from the input string.

**Syntax**
trim ( <string> )

**Parameters**
- <string> is a string.

**Returns**
String

**Example**
trim ( connection : \'DB''::'TBC.tbc.MARKET'.'STATE' )

**upper**
Converts all alphabetic characters in the input string to upper case.

**Syntax**
upper ( <string> )

**Parameters**
- <string> is a string.

**Returns**
String

**Example**
upper ( connection : \'DB''::'TBC.tbc.MARKET'.'STATE' )
### abs

Returns the absolute value of the input number.

**Syntax**

```
abs ( <number> )
```

**Parameters**

- `<number>` is a number.

**Returns**

Number

**Example**

```
abs ( connection : 'tbc'::'tbc.salesfact'.SALES )
```

### exp

Returns e(2.7182818) raised to the `arg`th power.

**Syntax**

```
exp ( <number> )
```

**Parameters**

- `<number>` is a number.

**Returns**

Number

**Example**

```
exp ( 3 )
```

### ln

Returns the natural logarithm of the input number.

**Syntax**

```
ln ( <number> )
```
Parameters

- <number> is a number.

Returns

Number

Example

\[ \ln(46) \]

**log10**

Returns the base 10 logarithm of the input number.

Syntax

\[ \log_{10}( <number> ) \]

Parameters

- <number> is a number.

Returns

Number

Example

\[ \log_{10}(136) \]

**pow**

Returns the value of \( x \) to the power of \( y \).

Syntax

\[ \text{pow}( <number_x> , <number_y> ) \]

Parameters

- <number_x> is a number.
- <number_y> is a number.

Returns

Number

Example

\[ \text{pow}(25,2) \]
**sqrt**

Returns the square root of the input number.

**Syntax**

```
sqrt ( <number> )
```

**Parameters**

- `<number>` is a number.

**Returns**

Number

**Example**

```
sqrt ( 16 )
```
! See bang character (!).

#MISSING See missing data (#MISSING).

access permissions A set of operations that a user can perform on a resource.

accessor Input and output data specifications for data mining algorithms.

account blocking The process by which accounts accept input data in the consolidated file. Blocked accounts do not receive their value through the additive consolidation process.

account eliminations Accounts which have their values set to zero in the consolidated file during consolidation.

account type How an account’s value flows over time, and its sign behavior. Account type options can include expense, income, asset, liability, and equity.

accountability map A visual, hierarchical representation of the responsibility, reporting, and dependency structure of the accountability teams (also known as critical business areas) in an organization.

accounts dimension A dimension type that makes accounting intelligence available. Only one dimension can be defined as Accounts.

active service A service whose Run Type is set to Start rather than Hold.

activity-level authorization Defines user access to applications and the types of activities they can perform on applications, independent of the data that will be operated on.

ad hoc report An online analytical query created on-the-fly by an end user.

adapter Software that enables a program to integrate with data and metadata from target and source systems.

adaptive states Interactive Reporting Web Client level of permission.

adjustment See journal entry (JE).

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.

agent An Essbase server process that starts and stops applications and databases, manages connections from users, and handles user-access security. The agent is referred to as ESSBASE.EXE.

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 function A type of function, such as sum or calculation of an average, that summarizes or performs analysis on data.

aggregate limit A limit placed on an aggregated request line item or aggregated metatopic item.

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 A collection of aggregate cells based on the levels of the members within each dimension. To reduce calculation time, values are pre-aggregated and stored as aggregate views. Retrievals start from aggregate view totals and add up from there.
aggregation  The process of rolling up and storing values in an aggregate storage database; the stored result of the aggregation process.

aggregation script  In aggregate storage databases only, a file that defines a selection of aggregate views to be built into an aggregation.

alias  An alternative name. For example, for a more easily identifiable column descriptor you can display the alias instead of the member name.

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.

ancestor  A branch member that has members below it. For example, the members Qtr2 and 2006 are ancestors of the member April.

appender  A Log4j term for destination.

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.

application currency  The default reporting currency for the application.

area  A predefined set of members and values that makes up a partition.

arithmetic data load  A data load that performs operations on values in the database, such as adding 10 to each value.

artifact  An individual application or repository item; for example, scripts, forms, rules files, Interactive Reporting documents, and financial reports. Also known as an object.

assemblies  Installation files for EPM System products or components.

asset account  An account type that stores values that represent a company’s assets.

assignment  The association of a source and destination in the allocation model that controls the direction of allocated costs or revenue flow within Profitability and Cost Management.

attribute  Characteristic 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.

attribute association  A relationship in a database outline whereby a member in an attribute dimension describes a characteristic of a member of its base dimension. For example, if product 100-10 has a grape flavor, the product 100-10 has the Flavor attribute association of grape. Thus, the 100-10 member of the Product dimension is associated with the Grape member of the Flavor attribute dimension.

Attribute Calculations dimension  A system-defined dimension that performs these calculation operations on groups of members: Sum, Count, Avg, Min, and Max. This dimension is calculated dynamically and is not visible in the database outline. For example, using the Avg member, you can calculate the average sales value for Red products in New York in January.

attribute dimension  A type of dimension that enables analysis based on the attributes or qualities of dimension members.

attribute reporting  A reporting process based on the attributes of the base dimension members. See also base dimension.

attribute type  A text, numeric, Boolean, date, or linked-attribute type that enables different functions for grouping, selecting, or calculating data. For example, because the Ounces attribute dimension has the type numeric, the number of ounces specified as the attribute of each product can be used to calculate the profit per ounce for that product.

authentication  Verification of identity as a security measure. Authentication is typically based on a user name and password. Passwords and digital signatures are forms of authentication.

authentication service  A core service that manages one authentication system.

auto-reversing journal  A journal for entering adjustments that you want to reverse in the next period.
automated stage  A stage that does not require human intervention, for example, a data load.

axis  (1) A straight line that passes through a graphic used for measurement and categorization.  (2) A report aspect used to arrange and relate multidimensional data, such as filters, pages, rows, and columns. For example, for a data query in Simple Basic, an axis can define columns for values for Qtr1, Qtr2, Qtr3, and Qtr4. Row data would be retrieved with totals in the following hierarchy: Market, Product.

backup  A duplicate copy of an application instance.

balance account  An account type that stores unsigned values that relate to a particular point in time.

balanced journal  A journal in which the total debits equal the total credits.

bang character  A character that terminates a series of report commands and requests information from the database. A report script must be terminated with a bang character; several bang characters can be used within a report script.

bar chart  A chart that can consist of one to 50 data sets, with any number of values assigned to each data set. Data sets are displayed as groups of corresponding bars, stacked bars, or individual bars in separate rows.

base currency  The currency in which daily business transactions are performed.

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.

base entity  An entity at the bottom of the organization structure that does not own other entities.

batch calculation  Any calculation on a database that is done in batch; for example, a calculation script or a full database calculation. Dynamic calculations are not considered to be batch calculations.

batch file  An operating system file that can call multiple ESSCMD scripts and run multiple sessions of ESSCMD. On Windows-based systems, batch files have BAT file extensions. On UNIX, batch files are written as a shell script.

batch loader  An FDM component that enables the processing of multiple files.

batch POV  A collection of all dimensions on the user POV of every report and book in the batch. While scheduling the batch, you can set the members selected on the batch POV.

batch processing mode  A method of using ESSCMD to write a batch or script file that can be used to automate routine server maintenance and diagnostic tasks. ESSCMD script files can execute multiple commands and can be run from the operating system command line or from within operating system batch files. Batch files can be used to call multiple ESSCMD scripts or run multiple instances of ESSCMD.

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.

Blocked Account  An account that you do not want calculated in the consolidated file because you want to enter it manually.

book  A container that holds a group of similar Financial Reporting documents. Books may specify dimension sections or dimension changes.

book POV  The dimension members for which a book is run.

bookmark  A link to a reporting document or a Web site, displayed on a personal page of a user. The two types of bookmarks are My Bookmarks and image bookmarks.

bounding rectangle  The required perimeter that encapsulates the Interactive Reporting document content when embedding Interactive Reporting document sections in a personal page, specified in pixels for height and width or row per page.

broadcast message  A simple text message sent by an administrator to a user who is logged on to a Planning application. The message displays information to the user such as system availability, notification of application refresh, or application backups.

budget administrator  A person responsible for setting up, configuring, maintaining, and controlling an application. Has all application privileges and data access permissions.
**build method** A method used to modify database outlines. Choice of a build method is based on the format of data in data source files.

**business process** A set of activities that collectively accomplish a business objective.

**business rules** Logical expressions or formulas that are created within an application to produce a desired set of resulting values.

**cache** A buffer in memory that holds data temporarily.

**calc script** A set of commands that define how a database is consolidated or aggregated. A calculation script may also contain commands that specify allocation and other calculation rules separate from the consolidation process.

**calculated member in MaxL DML** A member designed for analytical purposes and defined in the optional WITH section of a MaxL DML query.

**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.

**cell note** A text annotation for a cell in an Essbase database. Cell notes are a type of LRO.

**CHANGED status** Consolidation status that indicates data for an entity has changed.

**chart** A graphical representation of spreadsheet data. The visual nature expedites analysis, color-coding, and visual cues that aid comparisons.

**chart template** A template that defines the metrics to display in Workspace charts.

**child** A member with a parent above it in the database outline.

**choice list** A list of members that a report designer can specify for each dimension when defining the report’s point of view. A user who wants to change the point of view for a dimension that uses a choice list can select only the members specified in that defined member list or those members that meet the criteria defined in the function for the dynamic list.

**clean block** A data block that where the database is fully calculated, if a calculation script calculates all dimensions at once, or if the SET CLEARUPDATESTATUS command is used in a calculation script.

**cluster** An array of servers or databases that behave as a single resource which share task loads and provide failover support; eliminates one server or database as a single point of failure in a system.

**clustered bar charts** Charts in which categories are viewed side-by-side; useful for side-by-side category analysis; used only with vertical bar charts.

**categories** Groupings by which data is organized. For example, Month.

**cause and effect map** Depicts how the elements that form your corporate strategy relate and how they work together to meet your organization’s strategic goals. A Cause and Effect map tab is automatically created for each Strategy map.

**CDF** See custom-defined function (CDF).

**CDM** See custom-defined macro (CDM).

**calendar** User-defined time periods and their relationship to each other. Q1, Q2, Q3, and Q4 comprise a calendar or fiscal year.

**cascade** The process of creating multiple reports for a subset of member values.

**Catalog pane** Displays a list of elements available to the active section. If Query is the active section, a list of database tables is displayed. If Pivot is the active section, a list of results columns is displayed. If Dashboard is the active section, a list of embeddable sections, graphic tools, and control tools are displayed.
**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.

**committed access** An Essbase Kernel Isolation Level setting that affects how Essbase handles transactions. Under committed access, concurrent transactions hold long-term write locks and yield predictable results.

**computed item** A virtual column (as opposed to a column that is physically stored in the database or cube) that can be calculated by the database during a query, or by Interactive Reporting Studio in the Results section. Computed items are calculations of data based on functions, data items, and operators provided in the dialog box and can be included in reports or reused to calculate other data.

**configuration file** The security platform relies on XML documents to be configured by the product administrator or software installer. The XML document must be modified to indicate meaningful values for properties, specifying locations and attributes pertaining to the corporate authentication scenario.

**connection file** See Interactive Reporting connection file (.oce).

**consolidated file (Parent)** A file into which all of the business unit files are consolidated; contains the definition of the consolidation.

**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 file (*.cns)** The consolidation file is a graphical interface that enables you to add, delete or move Strategic Finance files in the consolidation process using either a Chart or Tree view. It also enables you to define and modify the consolidation.

**consolidation rule** Identifies the rule that is executed during the consolidation of the node of the hierarchy. This rule can contain customer specific formulas appropriate for the correct consolidation of parent balances. Elimination processing can be controlled within these rules.

**content** Information stored in the repository for any type of file.

**content browser** A Component that allows users to Browse and select content to be placed in a Workspace Page.

**context variable** A variable that is defined for a particular task flow to identify the context of the taskflow instance.

**contribution** The value added to a parent from a child entity. Each child has a contribution to its parent.

**controls group** Used in FDM to maintain and organize certification and assessment information, especially helpful for meeting Sarbanes-Oxley requirements.

**conversion rate** See exchange rate.

**cookie** A segment of data placed on your computer by a Web site.

**correlated subqueries** Subqueries that are evaluated once for every row in the parent query; created by joining a topic item in the subquery with a topic in the parent query.

**critical business area (CBA)** An individual or a group organized into a division, region, plant, cost center, profit center, project team, or process; also called accountability team or business area.

**critical success factor (CSF)** A capability that must be established and sustained to achieve a strategic objective; owned by a strategic objective or a critical process and is a parent to one or more actions.

**crosstab reporting** Categorizes and summarizes data in table format. The table cells contain summaries of the data that fit within the intersecting categories. For example, a crosstab report of product sales information could show size attributes, such as Small and Large, as column headings and color attributes, such as Blue and Yellow, as row headings. The cell in the table where Large and Blue intersect could contain the total sales of all Blue products that are sized Large.

**cube** A block of data that contains three or more dimensions. An Essbase database is a cube.
**cube deployment**  In Essbase Studio, the process of setting load options for a model to build an outline and load data into an Essbase application and database.

**cube schema**  In Essbase Studio, the metadata elements, such as measures and hierarchies, representing the logical model of a cube.

**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.

**Currency Overrides**  In any input period, the selected input method can be overridden to enable input of that period’s value as Default Currency/Items. To override the input method, enter a pound sign (#) either before or after the number.

**currency partition**  A dimension type that separates local currency members from a base currency, as defined in an application. Identifies currency types, such as Actual, Budget, and Forecast.

**custom calendar**  Any calendar created by an administrator.

**custom dimension**  A dimension created and defined by users. Channel, product, department, project, or region could be custom dimensions.

**custom property**  A property of a dimension or dimension member that is created by a user.

**custom report**  A complex report from the Design Report module, composed of any combination of components.

**custom-defined function (CDF)**  Essbase calculation functions developed in Java and added to the standard Essbase calculation scripting language using MaxL. See also custom-defined macro (CDM).

**custom-defined macro (CDM)**  Essbase macros written with Essbase calculator functions and special macro functions. Custom-defined macros use an internal Essbase macro language that enables the combination of calculation functions and they operate on multiple input parameters. See also custom-defined function (CDF).

**cycle through**  To perform multiple passes through a database while calculating it.

**dashboard**  A collection of metrics and indicators that provide an interactive summary of your business. Dashboards enable you to build and deploy analytic applications.

**data cache**  A buffer in memory that holds uncompressed data blocks.

**data cell**  See cell.

**data file cache**  A buffer in memory that holds compressed data (PAG) files.

**data form**  A grid display that enables users to enter data into the database from an interface such as a Web browser, and to view and analyze data or related text. Certain dimension member values are fixed, giving users a specific view into the data.

**data function**  That computes aggregate values, including averages, maximums, counts, and other statistics, that summarize groupings of data.

**data load**  The process of loading data to dimensions and members in Essbase outlines.

**data load location**  In FDM, a reporting unit responsible for submitting source data into the target system. Typically, there is one FDM data load location for each source file loaded to the target system.

**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 lock**  Prevents changes to data according to specified criteria, such as period or scenario.

**data mining**  The process of searching through an Essbase database for hidden relationships and patterns in a large amount of data.

**data model**  A representation of a subset of database tables.

**data value**  See cell.

**database connection**  File that stores definitions and properties used to connect to data sources and enables database references to be portable and widely used.
date measure  In Essbase, a member tagged as “Date” in the dimension where measures are represented. The cell values are displayed as formatted dates. Dates as measures can be useful for types of analysis that are difficult to represent using the Time dimension. For example, an application may need to track acquisition dates for a series of capital assets, but the acquisition dates span too large a period to allow for feasible Time dimension modeling. See also typed measure.

Default Currency Units  Define the unit scale of data. For example, if you select to define your analysis in Thousands, and enter “10”, this is interpreted as “10,000”.

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.

dependent entity  An entity that is owned by another entity in the organization.

derived text measure  In Essbase Studio, a text measure whose values are governed by a predefined rule expressed as a range. For example, a derived text measure, called "Sales Performance Index," based on a measure Sales, could consist of the values "High," "Medium," and "Low." This derived text measure is defined to display "High," "Medium," and "Low," depending on the range in which the corresponding sales values fall. See also text measure.

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.

Design Report  An interface in Web Analysis Studio for designing custom reports, from a library of components.

destination  Within a Profitability and Cost Management assignment, the destination is the receiving point for allocated values.

destination currency  The currency to which balances are converted. You enter exchange rates and convert from the source currency to the destination currency. For example, when you convert from EUR to USD, the destination currency is USD.

detail chart  A chart that provides the detailed information that you see in a Summary chart. Detail charts appear in the Investigate Section in columns below the Summary charts. If the Summary chart shows a Pie chart, then the Detail charts below represent each piece of the pie.

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 build  The process of adding dimensions and members to an Essbase outline.

dimension build rules  Specifications, similar to data load rules, that Essbase uses to modify an outline. The modification is based on data in an external data source file.

dimension tab  In the Pivot section, the tab that enables you to pivot data between rows and columns.

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.

dimensionality  In MaxL DML, the represented dimensions (and the order in which they are represented) in a set. For example, the following set consists of two tuples of the same dimensionality because they both reflect the dimensions (Region, Year): { (West, Feb), (East, Mar) }

direct rate  A currency rate that you enter in the exchange rate table. The direct rate is used for currency conversion. For example, to convert balances from JPY to USD, In the exchange rate table, enter a rate for the period/scenario where the source currency is JPY and the destination currency is USD.

dirty block  A data block containing cells that have been changed since the last calculation. Upper level blocks are marked as dirty if their child blocks are dirty (that is, they have been updated).

display type  One of three Web Analysis formats saved to the repository: spreadsheet, chart, and pinboard.
dog-ear The flipped page corner in the upper right corner of the chart header area.

domain In data mining, a variable representing a range of navigation within data.

drill-down Navigation through the query result set using the dimensional hierarchy. Drilling down moves the user perspective from aggregated data to detail. For example, drilling down can reveal hierarchical relationships between years and quarters or quarters and months.

drill-through The navigation from a value in one data source to corresponding data in another source.

driver A driver is an allocation method that describes the mathematical relationship between the sources that utilize the driver, and the destinations to which those sources allocate cost or revenue.

duplicate alias name A name that occurs more than once in an alias table and that can be associated with more than one member in a database outline. Duplicate alias names can be used with duplicate member outlines only.

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 hierarchy In aggregate storage database outlines only, a hierarchy in which members are calculated at retrieval time.

dynamic member list A system-created named member set that is based on user-defined criteria. The list is refreshed automatically whenever it is referenced in the application. As dimension members are added and deleted, the list automatically reapplies the criteria to reflect the changes.

dynamic reference A pointer in the rules file to header records in a data source.

dynamic report A report containing data that is updated when you run the report.

Dynamic Time Series A process that performs period-to-date reporting in block storage databases.

dynamic view account An account type indicating that account values are calculated dynamically from the data that is displayed.

Eliminated Account An account that does not appear in the consolidated file.

elimination The process of zeroing out (eliminating) transactions between entities within an organization.

employee A user responsible for, or associated with, specific business objects. Employees need not work for an organization; for example, they can be consultants. Employees must be associated with user accounts for authorization purposes.

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.

ending period A period enabling you to adjust the date range in a chart. For example, an ending period of “month”, produces a chart showing information through the end of the current month.

Enterprise View An Administration Services feature that enables management of the Essbase environment from a graphical tree view. From Enterprise View, you can operate directly on Essbase artifacts.

entity A dimension representing organizational units. Examples: divisions, subsidiaries, plants, regions, products, or other financial reporting units.
**Equity Beta** The riskiness of a stock, measured by the variance between its return and the market return, indicated by an index called “beta”. For example, if a stock’s return normally moves up or down 1.2% when the market moves up or down 1%, the stock has a beta of 1.2.

**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.

**EssCell** A function entered into a cell in Essbase Spreadsheet Add-in to retrieve a value representing an intersection of specific Essbase database members.

**ESSCMD** A command-line interface for performing Essbase operations interactively or through batch script files.

**ESSLANG** The Essbase environment variable that defines the encoding used to interpret text characters. See also encoding.

**ESSMSH** See MaxL Shell.

**exceptions** Values that satisfy predefined conditions. You can define formatting indicators or notify subscribing users when exceptions are generated.

**exchange rate** A numeric value for converting one currency to another. For example, to convert 1 USD into EUR, the exchange rate of 0.8936 is multiplied with the U.S. dollar. The European euro equivalent of $1 is 0.8936.

**exchange rate type** An identifier for an exchange rate. Different rate types are used because there may be multiple rates for a period and year. Users traditionally define rates at period end for the average rate of the period and for the end of the period. Additional rate types are historical rates, budget rates, forecast rates, and so on. A rate type applies to one point in time.

**expense account** An account that stores periodic and year-to-date values that decrease net worth if they are positive.

**Extensible Markup Language (XML)** A language comprising a set of tags used to assign attributes to data that can be interpreted between applications according to a schema.

**external authentication** Logging on to Oracle’s Hyperion applications with user information stored outside the applications, typically in a corporate directory such as MSAD or NTLM.

**externally triggered events** Non-time-based events for scheduling job runs.

**Extract, Transform, and Load (ETL)** Data source-specific programs for extracting data and migrating it to applications.

**extraction command** An Essbase reporting command that handles the selection, orientation, grouping, and ordering of raw data extracted from a database; begins with the less than (<) character.

**fact table** The central table in a star join schema, characterized by a foreign key and elements drawn from a dimension table. This table typically contains numeric data that can be related to all other tables in the schema.

**Favorites gadget** Contains links to Reporting and Analysis documents and URLs.

**field** An item in a data source file to be loaded into an Essbase database.

**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.

**flow account** An unsigned account that stores periodic and year-to-date values.

**folder** A file containing other files for the purpose of structuring a hierarchy.

**footer** Text or images at the bottom of report pages, containing dynamic functions or static text such as page numbers, dates, logos, titles or file names, and author names.

**format** Visual characteristics of documents or report objects.

**format string** In Essbase, a method for transforming the way cell values are displayed.

**formula** A combination of operators, functions, dimension and member names, and numeric constants calculating database members.

**frame** An area on the desktop. There are two main areas: the navigation and Workspace frames.
**free-form grid**  An object for presenting, entering, and integrating data from different sources for dynamic calculations.

**free-form reporting**  Creating reports by entering dimension members or report script commands in worksheets.

**function**  A routine that returns values or database members.

**gadget**  Simple, specialized, lightweight applications that provide easy viewing of EPM content and enable access to core Reporting and Analysis functionality.

**genealogy data**  Additional data that is optionally generated after allocation calculations. This data enables reporting on all cost or revenue flows from start to finish through all allocation steps.

**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. Use the unique generation name to identify a layer in the hierarchical tree structure.

**generic jobs**  Non-SQR Production Reporting or non-Interactive Reporting jobs.

**global report command**  A command in a running report script that is effective until replaced by another global command or the file ends.

**grid POV**  A means for specifying dimension members on a grid without placing dimensions in rows, columns, or page intersections. A report designer can set POV values at the grid level, preventing user POVs from affecting the grid. If a dimension has one grid value, you put the dimension into the grid POVs instead of the row, column, or page.

**group**  A container for assigning similar access permissions to multiple users.

**GUI**  Graphical user interface

**head up display**  A mode that shows your loaded Smart Space desktop including the background image above your Windows desktop.

**highlighting**  Depending on your configuration, chart cells or ZoomChart details may be highlighted, indicating value status: red (bad), yellow (warning), or green (good).

**Historical Average**  An average for an account over a number of historical periods.

**holding company**  An entity that is part of a legal entity group, with direct or indirect investments in all entities in the group.

**host**  A server on which applications and services are installed.

**host properties**  Properties pertaining to a host, or if the host has multiple Install_Homes, to an Install_Home. The host properties are configured from the CMC.

**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.

**hyperlink**  A link to a file, Web page, or an intranet HTML page.

**Hypertext Markup Language (HTML)**  A programming language specifying how Web browsers display data.

**identity**  A unique identification for a user or group in external authentication.

**image bookmarks**  Graphic links to Web pages or repository items.

**IMPACTED status**  Indicates changes in child entities consolidating into parent entities.

**implied share**  A member with one or more children, but only one is consolidated, so the parent and child share a value.

**import format**  In FDM, defines the structure of the source file which enables the loading of a source data file to an FDM data load location.

**inactive group**  A group for which an administrator has deactivated system access.

**inactive service**  A service suspended from operating.

**INACTIVE status**  Indicates entities deactivated from consolidation for the current period.

**inactive user**  A user whose account has been deactivated by an administrator.

**income account**  An account storing periodic and year-to-date values that, if positive, increase net worth.

**index**  (1) A method where Essbase uses sparse-data combinations to retrieve data in block storage databases. (2) The index file.
index cache  A buffer containing index pages.

index entry  A pointer to an intersection of sparse dimensions. Index entries point to data blocks on disk and use offsets to locate cells.

index file  An Essbase file storing block storage data retrieval information, residing on disk, and containing index pages.

index page  A subdivision in an index file. Contains pointers to data blocks.

input data  Data loaded from a source rather than calculated.

Install_Home  A variable for the directory where EPM System products are installed. Refers to one instance of an EPM System product when multiple applications are installed on the same computer.

integration  Process that is run to move data between EPM System products using Shared Services. Data integration definitions specify the data moving between a source application and a destination application, and enable the data movements to be grouped, ordered, and scheduled.

intelligent calculation  A calculation method tracking updated data blocks since the last calculation.

Interactive Reporting connection file (.oce)  Files encapsulating database connection information, including: the database API (ODBC, SQL*Net, etc.), database software, the database server network address, and database user name. Administrators create and publish Interactive Reporting connection files (.oce).

intercompany elimination  See elimination.

intercompany matching  The process of comparing balances for pairs of intercompany accounts within an application. Intercompany receivables are compared to intercompany payables for matches. Matching accounts are used to eliminate intercompany transactions from an organization’s consolidated totals.

intercompany matching report  A report that compares intercompany account balances and indicates if the accounts are in, or out, of balance.

interdimensional irrelevance  A situation in which a dimension does not intersect with other dimensions. Because the data in the dimension cannot be accessed from the non-intersecting dimensions, the non-intersecting dimensions are not relevant to that dimension.

intersection  A unit of data representing the intersection of dimensions in a multidimensional database; also, a worksheet cell.

intrastage assignment  Assignments in the financial flow that are assigned to objects within the same stage.

introspection  A deep inspection of a data source to discover hierarchies based on the inherent relationships in the database. Contrast with scraping.

Investigation  See drill-through.

isolation level  An Essbase Kernel setting that determines the lock and commit behavior of database operations. Choices are: committed access and uncommitted access.

iteration  A “pass” of the budget or planning cycle in which the same version of data is revised and promoted.

Java Database Connectivity (JDBC)  A client-server communication protocol used by Java based clients and relational databases. The JDBC interface provides a call-level API for SQL-based database access.

job output  Files or reports produced from running a job.

jobs  Documents with special properties that can be launched to generate output. A job can contain Interactive Reporting, SQR Production Reporting, or generic documents.

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.

journal entry (JE)  A set of debit/credit adjustments to account balances for a scenario and period.

JSP  Java Server Pages.

KeyContacts gadget  Contains a group of Smart Space users and provides access to Smart Space Collaborator. For example, you can have a KeyContacts gadget for your marketing team and another for your development team.

latest  A Spreadsheet key word used to extract data values from the member defined as the latest time period.
layer  (1) The horizontal location of members in a hierarchical structure, specified by generation (top down) or level (bottom up). (2) Position of objects relative to other objects. For example, in the Sample Basic database, Qtr1 and Qtr4 are in the same layer, so they are also in the same generation, but in a database with a ragged hierarchy, Qtr1 and Qtr4 might not be in the same layer, though they are in the same generation.

layout area Used to designate an area on a Workspace Page where content can be placed.

legend box A box containing labels that identify the data categories of a dimension.

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.

liability account An account type that stores “point in time” balances of a company’s liabilities. Examples of liability accounts include accrued expenses, accounts payable, and long term debt.

life cycle management The process of managing application information from inception to retirement.

Lifecycle Management Utility A command-line utility for migrating applications and artifacts.

line chart A chart that displays one to 50 data sets, each represented by a line. A line chart can display each line stacked on the preceding ones, as represented by an absolute value or a percent.

line item detail The lowest level of detail in an account.

lineage The relationship between different metadata elements showing how one metadata element is derived from one or more other metadata elements, ultimately tracing the metadata element to its physical source. In Essbase Studio, a lineage viewer displays the relationships graphically. See also traceability.

link  (1) A reference to a repository object. Links can reference folders, files, shortcuts, and other links. (2) In a task flow, the point where the activity in one stage ends and another begins.

link condition A logical expression evaluated by the taskflow engine to determine the sequence of launching taskflow stages.

linked data model Documents that are linked to a master copy in a repository.

linked partition A shared partition that enables you to use a data cell to link two databases. When a user clicks a linked cell in a worksheet, Essbase opens a new sheet displaying the dimensions in the linked database. The user can then drill down those dimensions.

linked reporting object (LRO) A cell-based link to an external file such as cell notes, URLs, or files with text, audio, video, or pictures. (Only cell notes are supported for Essbase LROs in Financial Reporting.) Contrast with local report object.

local currency An input currency type. When an input currency type is not specified, the local currency matches the entity’s base currency.

local report object A report object that is not linked to a Financial Reporting report object in Explorer. Contrast with linked reporting object (LRO).

local results A data model’s query results. Results can be used in local joins by dragging them into the data model. Local results are displayed in the catalog when requested.

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.

locale header record A text record at the beginning of some non-Unicode-encoded text files, such as scripts, that identifies the encoding locale.

location alias A descriptor that identifies a data source. The location alias specifies a server, application, database, user name, and password. Location aliases are set by DBAs at the database level using Administration Services Console, ESSCMD, or the API.

locked A user-invoked process that prevents users and processes from modifying data.
locked data model  Data models that cannot be modified by a user.

LOCKED status  A consolidation status indicating that an entity contains data that cannot be modified.

Log Analyzer  An Administration Services feature that enables filtering, searching, and analysis of Essbase logs.

logic group  In FDM, contains one or more logic accounts that are generated after a source file is loaded into FDM. Logic accounts are calculated accounts that are derived from the source data.

LRO  See linked reporting object (LRO).

managed server  An application server process running in its own Java Virtual Machine (JVM).

manual stage  A stage that requires human intervention to complete.

Map File  Used to store the definition for sending data to or retrieving data from an external database. Map files have different extensions (.mps to send data; .mpr to retrieve data).

Map Navigator  A feature that displays your current position on a Strategy, Accountability, or Cause and Effect map, indicated by a red outline.

Marginal Tax Rate  Used to calculate the after-tax cost of debt. Represents the tax rate applied to the last earned income dollar (the rate from the highest tax bracket into which income falls) and includes federal, state and local taxes. Based on current level of taxable income and tax bracket, you can predict marginal tax rate.

Market Risk Premium  The additional rate of return paid over the risk-free rate to persuade investors to hold “riskier” investments than government securities. Calculated by subtracting the risk-free rate from the expected market return. These figures should closely model future market conditions.

master data model  An independent data model that is referenced as a source by multiple queries. When used, “Locked Data Model” is displayed in the Query section’s Content pane; the data model is linked to the master data model displayed in the Data Model section, which an administrator may hide.

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 %.

MaxL  The multidimensional database access language for Essbase, consisting of a data definition language (MaxL DDL) and a data manipulation language (MaxL DML). See also MaxL DDL, MaxL DML, and MaxL Shell.

MaxL DDL  Data definition language used by Essbase for batch or interactive system-administration tasks.

MaxL DML  Data manipulation language used in Essbase for data query and extraction.

MaxL Perl Module  A Perl module (essbase.pm) that is part of Essbase MaxL DDL. This module can be added to the Perl package to provide access to Essbase databases from Perl programs.

MaxL Script Editor  A script-development environment in Administration Services Console. MaxL Script Editor is an alternative to using a text editor and the MaxL Shell for administering Essbase with MaxL scripts.

MaxL Shell  An interface for passing MaxL statements to Essbase Server. The MaxL Shell executable file is located in the Essbase bin directory (UNIX: essmsh, Windows: essmsh.exe).

MDX (multidimensional expression)  The language that give instructions to OLE DB for OLAP- compliant databases, as SQL is used for relational databases. When you build the OLAPQuery section’s Outliner, Interactive Reporting Clients translate requests into MDX instructions. When you process the query, MDX is sent to the database server, which returns records that answer your query. See also SQL spreadsheet.

measures  Numeric values in an OLAP database cube that are available for analysis. Measures are margin, cost of goods sold, unit sales, budget amount, and so on. See also fact table.

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 list  A named group, system- or user-defined, that references members, functions, or member lists within a dimension.

member load  The process of adding dimensions and members (without data) to Essbase outlines.

member selection report command  A type of Report Writer command that selects member ranges based on outline relationships, such as sibling, generation, and level.

member-specific report command  A type of Report Writer formatting command that is executed as it is encountered in a report script. The command affects only its associated member and executes the format command before processing the member.

merge  A data load option that clears values only from the accounts specified in the data load file and replaces them with values in the data load file.

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.

metadata elements  Metadata derived from data sources and other metadata that is stored and cataloged for Essbase Studio use.

metadata sampling  The process of retrieving a sample of members in a dimension in a drill-down operation.

metadata security  Security set at the member level to restrict users from accessing certain outline members.

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

metric  A numeric measurement computed from business data to help assess business performance and analyze company trends.

migration  The process of copying applications, artifacts, or users from one environment or computer to another; for example, from a testing environment to a production environment.

migration audit report  A report generated from the migration log that provides tracking information for an application migration.

migration definition file (.mdf)  A file that contains migration parameters for an application migration, enabling batch script processing.

migration log  A log file that captures all application migration actions and messages.

migration snapshot  A snapshot of an application migration that is captured in the migration log.

MIME Type  (Multipurpose Internet Mail Extension) An attribute that describes the data format of an item, so that the system knows which application should open the object. A file’s mime type is determined by the file extension or HTTP header. Plug-ins tell browsers what mime types they support and what file extensions correspond to each mime type.

mining attribute  In data mining, a class of values used as a factor in analysis of a set of data.

minireport  A report component that includes layout, content, hyperlinks, and the query or queries to load the report. Each report can include one or more minireports.

minischema  A graphical representation of a subset of tables from a data source that represents a data modeling context.

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.

model  (1) In data mining, a collection of an algorithm’s findings about examined data. A model can be applied against a wider data set to generate useful information about that data. (2) A file or content string containing an application-specific representation of data. Models are the basic data managed by Shared Services, of two major types: dimensional and non-dimensional application objects. (3) In Business Modeling, a network of boxes connected to represent and calculate the operational and financial flow through the area being examined.

monetary  A money-related value.

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. Contrast with relational database.
multiload  An FDM feature that allows the simultaneous loading of multiple periods, categories, and locations.

My Workspace Page  A page created with content from multiple sources including documents, URL, and other content types. Enables a user to aggregate content from Oracle and non-Oracle sources.

named set  In MaxL DML, a set with its logic defined in the optional WITH section of a MaxL DML query. The named set can be referenced multiple times in the query.

native authentication  The process of authenticating a user name and password from within the server or application.

nested column headings  A report column heading format that displays data from multiple dimensions. For example, a column heading that contains Year and Scenario members is a nested column. The nested column heading shows Q1 (from the Year dimension) in the top line of the heading, qualified by Actual and Budget (from the Scenario dimension) in the bottom line of the heading.

NO DATA status  A consolidation status indicating that this entity contains no data for the specified period and account.

non-dimensional model  A Shared Services model type that includes application objects such as security files, member lists, calculation scripts, and Web forms.

non-unique member name  See duplicate member name.

note  Additional information associated with a box, measure, scorecard or map element.

Notifications gadget  Shows notification message history received from other users or systems.

null value  A value that is absent of data. Null values are not equal to zero.

numeric attribute range  A feature used to associate a base dimension member that has a discrete numeric value with an attribute that represents a value range. For example, to classify customers by age, an Age Group attribute dimension can contain members for the following age ranges: 0-20, 21-40, 41-60, and 61-80. Each Customer dimension member can be associated with an Age Group range. Data can be retrieved based on the age ranges rather than on individual age values.

ODBC  Open Database Connectivity. A database access method used from any application regardless of how the database management system (DBMS) processes the information.

OK status  A consolidation status indicating that an entity has already been consolidated, and that data has not changed below it in the organization structure.

OLAP Metadata Catalog  In 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 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.

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.

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

organization  An entity hierarchy that defines each entity and their relationship to others in the hierarchy.

origin  The intersection of two axes.

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.

outline synchronization  For partitioned databases, the process of propagating outline changes from one database to another database.

P&L accounts (P&L)  Profit and loss accounts. Refers to a typical grouping of expense and income accounts that comprise a company's income statement.

page  A display of information in a grid or table often represented by the Z-axis. A page can contain data from one field, derived data from a calculation, or text.

page file  Essbase data file.
page heading  A report heading type that lists members represented on the current page of the report. All data values on the page have the members in the page heading as a common attribute.

page member  A member that determines the page axis.

palette  A JASC compliant file with a .PAL extension. Each palette contains 16 colors that complement each other and can be used to set the dashboard color elements.

parallel calculation  A calculation option. Essbase divides a calculation into tasks and calculates some tasks simultaneously.

parallel data load  In Essbase, the concurrent execution of data load stages by multiple process threads.

parallel export  The ability to export Essbase data to multiple files. This may be faster than exporting to a single file, and it may resolve problems caused by a single data file becoming too large for the operating system to handle.

parent adjustments  The journal entries that are posted to a child in relation to its parent.

parents  The entities that contain one or more dependent entities that report directly to them. Because parents are both entities and associated with at least one node, they have entity, node, and parent information associated with them.

partition area  A sub cube within a database. A partition is composed of one or more areas of cells from a portion of the database. For replicated and transparent partitions, the number of cells within an area must be the same for the data source and target to ensure that the two partitions have the same shape. If the data source area contains 18 cells, the data target area must also contain 18 cells to accommodate the number of values.

partitioning  The process of defining areas of data that are shared or linked between data models. Partitioning can affect the performance and scalability of Essbase applications.

pattern matching  The ability to match a value with any or all characters of an item entered as a criterion. Missing characters may be represented by wild card values such as a question mark (?) or an asterisk (*). For example, “Find all instances of apple” returns apple, but “Find all instances of apple*” returns apple, applesauce, applecranberry, and so on.

percent consolidation  The portion of a child’s values that is consolidated to its parent.

percent control  Identifies the extent to which an entity is controlled within the context of its group.

percent ownership  Identifies the extent to which an entity is owned by its parent.

performance indicator  An image file used to represent measure and scorecard performance based on a range you specify; also called a status symbol. You can use the default performance indicators or create an unlimited number of your own.

periodic value method (PVA)  A process of currency conversion that applies the periodic exchange rate values over time to derive converted results.

permission  A level of access granted to users and groups for managing data or other users and groups.

persistence  The continuance or longevity of effect for any Essbase operation or setting. For example, an Essbase administrator may limit the persistence of user name and password validity.

personal pages  A personal window to repository information. You select what information to display and its layout and colors.

personal recurring time events  Reusable time events that are accessible only to the user who created them.

personal variable  A named selection statement of complex member selections.

perspective  A category used to group measures on a scorecard or strategic objectives within an application. A perspective can represent a key stakeholder (such as a customer, employee, or shareholder/financial) or a key competency area (such as time, cost, or quality).

pie chart  A chart that shows one data set segmented in a pie formation.

pinboard  One of the three data object display types. Pinboards are graphics, composed of backgrounds and interactive icons called pins. Pinboards require traffic lighting definitions.
pins Interactive icons placed on graphic reports called pinboards. Pins are dynamic. They can change images and traffic lighting color based on the underlying data values and analysis tools criteria.

pivot The ability to alter the perspective of retrieved data. When Essbase first retrieves a dimension, it expands data into rows. You can then pivot or rearrange the data to obtain a different viewpoint.

planner Planners, who comprise the majority of users, can input and submit data, use reports that others create, execute business rules, use task lists, enable e-mail notification for themselves, and use Smart View.

planning unit A data slice at the intersection of a scenario, version, and entity; the basic unit for preparing, reviewing, annotating, and approving plan data.

plot area The area bounded by X, Y, and Z axes; for pie charts, the rectangular area surrounding the pie.

plug account An account in which the system stores any out of balance differences between intercompany account pairs during the elimination process.

post stage assignment Assignments in the allocation model that are assigned to locations in a subsequent model stage.

POV (point of view) A feature for setting data focus by selecting members that are not already assigned to row, column, or page axes. For example, selectable POVs in FDM could include location, period, category, and target category. In another example, using POV as a filter in Smart View, you could assign the Currency dimension to the POV and select the Euro member. Selecting this POV in data forms displays data in Euro values.

precalkulation Calculating the database prior to user retrieval.

precision Number of decimal places displayed in numbers.

predefined drill paths Paths used to drill to the next level of detail, as defined in the data model.

presentation A playlist of Web Analysis documents, enabling reports to be grouped, organized, ordered, distributed, and reviewed. Includes pointers referencing reports in the repository.

preserve formulas User-created formulas kept within a worksheet while retrieving data.

primary measure A high-priority measure important to your company and business needs. Displayed in the Contents frame.

process monitor report Displays a list of locations and their positions within the FDM data conversion process. You can use the process monitor report to monitor the status of the closing process. The report is time-stamped. Therefore, it can be used to determine to which locations at which time data was loaded.

product In Shared Services, an application type, such as Planning or Performance Scorecard.

Production Reporting See SQR Production Reporting.

project An instance of EPM System products grouped together in an implementation. For example, a Planning project may consist of a Planning application, an Essbase cube, and a Financial Reporting server instance.

property A characteristic of an artifact, such as size, type, or processing instructions.

provisioning The process of granting users and groups specific access permissions to resources.

proxy server A server acting as an intermediary between workstation users and the Internet to ensure security.

public job parameters Reusable, named job parameters created by administrators and accessible to users with requisite access privileges.

public recurring time events Reusable time events created by administrators and accessible through the access control system.

PVA See periodic value method (PVA).

qualified name A member name in a qualified format that differentiates duplicate member names in a duplicate member outline. For example, [Market].[East].[State]. [New York] or [Market].[East].[City].[New York]

query Information requests from data providers. For example, used to access relational data sources.

query governor An Essbase Integration server parameter or Essbase server configuration setting that controls the duration and size of queries made to data sources.
range  A set of values including upper and lower limits, and values falling between limits. Can contain numbers, amounts, or dates.

reciprocal assignment  An assignment in the financial flow that also has the source as one of its destinations.

reconfigure URL  URL used to reload servlet configuration settings dynamically when users are already logged on to the Workspace.

record  In a database, a group of fields making up one complete entry. For example, a customer record may contain fields for name, address, telephone number, and sales data.

recurring template  A journal template for making identical adjustments in every period.

recurring time event  An event specifying a starting point and the frequency for running a job.

redundant data  Duplicate data blocks that Essbase retains during transactions until Essbase commits updated blocks.

regular journal  A feature for entering one-time adjustments for a period. Can be balanced, balanced by entity, or unbalanced.

Related Accounts  The account structure groups all main and related accounts under the same main account number. The main account is distinguished from related accounts by the first suffix of the account number.

relational database  A type of database that stores data in related two-dimensional tables. Contrast with multidimensional database.

replace  A data load option that clears existing values from all accounts for periods specified in the data load file, and loads values from the data load file. If an account is not specified in the load file, its values for the specified periods are cleared.

replicated partition  A portion of a database, defined through Partition Manager, used to propagate an update to data mastered at one site to a copy of data stored at another site. Users can access the data as though it were part of their local database.

Report Extractor  An Essbase component that retrieves report data from the Essbase database when report scripts are run.

report object  In report designs, a basic element with properties defining behavior or appearance, such as text boxes, grids, images, and charts.

report script  A text file containing Essbase Report Writer commands that generate one or more production reports.

Report Viewer  An Essbase component that displays complete reports after report scripts are run.

reporting currency  The currency used to prepare financial statements, and converted from local currencies to reporting currencies.

repository  Stores metadata, formatting, and annotation information for views and queries.

resources  Objects or services managed by the system, such as roles, users, groups, files, and jobs.

restore  An operation to reload data and structural information after a database has been damaged or destroyed, typically performed after shutting down and restarting the database.

restructure  An operation to regenerate or rebuild the database index and, in some cases, data files.

result frequency  The algorithm used to create a set of dates to collect and display results.

review level  A Process Management review status indicator representing the process unit level, such as Not Started, First Pass, Submitted, Approved, and Published.

Risk Free Rate  The rate of return expected from “safer” investments such as long-term U.S. government securities.

role  The means by which access permissions are granted to users and groups for resources.

roll-up  See consolidation.

root member  The highest member in a dimension branch.

RSC services  Services that are configured with Remote Service Configurator, including Repository Service, Service Broker, Name Service, Event Service, and Job Service.

runtime prompt  A variable that users enter or select before a business rule is run.

sampling  The process of selecting a representative portion of an entity to determine the entity’s characteristics. See also metadata sampling.
**saved assumptions**  User-defined Planning assumptions that drive key business calculations (for example, the cost per square foot of office floor space).

**scaling**  Scaling determines the display of values in whole numbers, tens, hundreds, thousands, millions, and so on.

**scenario**  A dimension for classifying data (for example, Actuals, Budget, Forecast1, and Forecast2).

**scope**  The area of data encompassed by any Essbase operation or setting; for example, the area of data affected by a security setting. Most commonly, scope refers to three levels of granularity, where higher levels encompass lower levels. From highest to lowest, these levels are as follows: the entire system (Essbase Server), applications on Essbase servers, or databases within Essbase server applications. See also **persistence**.

**score**  The level at which targets are achieved, usually expressed as a percentage of the target.

**scorecard**  Business object that represents the progress of an employee, strategy element, or accountability element toward goals. Scorecards ascertain this progress based on data collected for each measure and child scorecard added to the scorecard.

**scraping**  An inspection of a data source to derive the most basic metadata elements from it. *Contrast with introspection.*

**Search gadget**  Searches the Reporting and Analysis repository. The Search gadget looks for a match in the document keywords and description, which are set when you import a document.

**secondary measure**  A low-priority measure, less important than primary measures. Secondary measures do not have Performance reports but can be used on scorecards and to create dimension measure templates.

**security agent**  A Web access management provider (for example, Netegrity SiteMinder) that protects corporate Web resources.

**security platform**  A framework enabling EPM System products to use external authentication and single sign-on.

**serial calculation**  The default calculation setting. Divides a calculation pass into tasks and calculates one task at a time.

**services**  Resources that enable business items to be retrieved, changed, added, or deleted. Examples: Authorization and Authentication.

**servlet**  A piece of compiled code executable by a Web server.

**Servlet Configurator**  A utility for configuring all locally installed servlets.

**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.

**Shared Services Registry**  Part of the Shared Services database, the Shared Services Registry stores and reuses information for most installed EPM System products, including installation directories, database settings, deployment settings, computer names, ports, servers, URLs, and dependent service data.

**Shared Workspace Page**  Workspace Pages shared across an organization which are stored in a special System folder and can be accessed by authorized users from the Shared Workspace Pages Navigate menu.

**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.

**single sign-on**  Ability to access multiple EPM System products after a single login using external credentials.

**smart slice**  In Smart View, a reusable perspective of a data source that contains a restricted set of dimensions or dimension members.

**Smart Space client software**  Runs on the client’s computer and provides gadgets, instant collaboration and access to the Reporting and Analysis repository. It is composed of the Smart Space framework and gadgets.

**Smart Space Collaborator**  A service that enables users or systems to send messages and share Reporting and Analysis repository content. The message can take many forms, including instant message style discussions, meetings, and toast messages.
**smart tags**  Keywords in Microsoft Office applications that are associated with predefined actions available from the Smart Tag menu. In EPM System products, smart tags can also be used to import Reporting and Analysis content, and access Financial Management and Essbase functions.

**SmartBook gadget**  Contains documents from the Reporting and Analysis repository or URLs. All documents are loaded when the SmartBook is opened so you can access all content immediately.

**SmartCut**  A link to a repository item, in URL form.

**snapshot**  Read-only data from a specific time.

**source currency**  The currency from which values originate and are converted through exchange rates to the destination currency.

**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. *Contrast with dense dimension.*

**SPF files**  Printer-independent files created by an SQR Production Reporting server, containing a representation of the actual formatted report output, including fonts, spacing, headers, footers, and so on.

**Spotlighter**  A tool that enables color coding based on selected conditions.

**SQL spreadsheet**  A data object that displays the result set of a SQL query.

**SQR Production Reporting**  A specialized programming language for data access, data manipulation, and creating SQR Production Reporting documents.

**stage**  A task description that forms one logical step within a taskflow, usually performed by an individual. A stage can be manual or automated.

**stage action**  For automated stages, the invoked action that executes the stage.

**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.

**standard journal template**  A journal function used to post adjustments that have common adjustment information for each period. For example, you can create a standard template that contains the common account IDs, entity IDs, or amounts, then use the template as the basis for many regular journals.

**Status bar**  The status bar at the bottom of the screen displays helpful information about commands, accounts, and the current status of your data file.

**stored hierarchy**  In aggregate storage databases outlines only. A hierarchy in which the members are aggregated according to the outline structure. Stored hierarchy members have certain restrictions, for example, they cannot contain formulas.

**strategic objective (SO)**  A long-term goal defined by measurable results. Each strategic objective is associated with one perspective in the application, has one parent, the entity, and is a parent to critical success factors or other strategic objectives.

**Strategy map**  Represents how the organization implements high-level mission and vision statements into lower-level, constituent strategic goals and objectives.

**structure view**  Displays a topic as a simple list of component data items.

**Structured Query Language**  A language used to process instructions to relational databases.

**Subaccount Numbering**  A system for numbering subaccounts using non-sequential, whole numbers.

**subscribe**  Flags an item or folder to receive automatic notification whenever the item or folder is updated.

**Summary chart**  In the Investigates Section, rolls up detail charts shown below in the same column, plotting metrics at the summary level at the top of each chart column.

**super service**  A special service used by the startCommonServices script to start the RSC services.

**supervisor**  A user with full access to all applications, databases, related files, and security mechanisms for a server.

**supporting detail**  Calculations and assumptions from which the values of cells are derived.
**suppress rows** Excludes rows containing missing values, and underscores characters from spreadsheet reports.

**symmetric multiprocessing (SMP)** A server architecture that enables multiprocessing and multithreading. Performance is not significantly degraded when a large number of users connect to a single instance simultaneously.

**sync** Synchronizes Shared Services and application models.

**synchronized** The condition that exists when the latest version of a model resides in both the application and in Shared Services. See also model.

**system extract** Transfers data from an application’s metadata into an ASCII file.

**tabs** Navigable views of accounts and reports in Strategic Finance.

**target** Expected results of a measure for a specified period of time (day, quarter, and so on).

**task list** A detailed status list of tasks for a particular user.

**taskflow** The automation of a business process in which tasks are passed from one taskflow participant to another according to procedural rules.

**taskflow definition** Represents business processes in the taskflow management system. Consists of a network of stages and their relationships; criteria indicating the start and end of the taskflow; and information about individual stages, such as participants, associated applications, associated activities, and so on.

**taskflow instance** Represents a single instance of a taskflow including its state and associated data.

**taskflow management system** Defines, creates, and manages the execution of a taskflow including: definitions, user or application interactions, and application executables.

**taskflow participant** The resource who performs the task associated with the taskflow stage instance for both manual and automated stages.

**Taxes - Initial Balances** Strategic Finance assumes that the Initial Loss Balance, Initial Gain Balance and the Initial Balance of Taxes Paid entries have taken place in the period before the first Strategic Finance time period.


**template** A predefined format designed to retrieve particular data consistently.

**text list** In Essbase, an object that stores text values mapped to numeric identifiers. Text Lists enable the use of text measures.

**text measure** A data type that allows measure values to be expressed as text. In Essbase, a member tagged as “Text” in the dimension where measures are represented. The cell values are displayed as predefined text. For example, the text measure “Satisfaction Index” may have the values Low, Medium, and High. See also typed measure, text list, derived text measure.

**time dimension** Defines the time period that the data represents, such as fiscal or calendar periods.

**time events** Triggers for execution of jobs.

**time line viewer** An FDM feature that allows a user to view dates and times of completed process flow steps for specific locations.

**time scale** Displays metrics by a specific period in time, such as monthly or quarterly.

**time series reporting** A process for reporting data based on a calendar date (for example, year, quarter, month, or week).

**Title bar** Displays the Strategic Finance name, the file name, and the scenario name Version box.

**toast message** Messages that appear in the lower right corner of the screen and fade in and out.

**token** An encrypted identification of one valid user or group on an external authentication system.

**top and side labels** Column and row headings on the top and sides of a Pivot report.

**top-level member** A dimension member at the top of the tree in a dimension outline hierarchy, or the first member of the dimension in sort order if there is no hierarchical relationship among dimension members. The top-level member name is generally the same as the dimension name if a hierarchical relationship exists.

**trace allocations** A feature of Profitability and Cost Management that enables you to visually follow the flow of financial data, either forwards or backwards, from a single intersection throughout the model.
trace level  Defines the level of detail captured in the log file.

traceability  The ability to track a metadata element to its physical source. For example, in Essbase Studio, a cube schema can be traced from its hierarchies and measure hierarchies, to its dimension elements, date/time elements, and measures, and ultimately, to its physical source elements.

traffic lighting  Color-coding of report cells, or pins based on a comparison of two dimension members, or on fixed limits.

transformation  (1) Transforms artifacts so that they function properly in the destination environment after application migration. (2) In data mining, modifies data (bidirectionally) flowing between the cells in the cube and the algorithm.

translation  See currency conversion.

Transmission Control Protocol/Internet Protocol (TCP/IP)  A standard set of communication protocols linking computers with different operating systems and internal architectures. TCP/IP utilities are used to exchange files, send mail, and store data to various computers that are connected to local and wide area networks.

transparent login  Logs in authenticated users without launching the login screen.

transparent partition  A shared partition that enables users to access and change data in a remote database as though it is part of a local database.

triangulation  A means of converting balances from one currency to another via a third common currency. In Europe, this is the euro for member countries. For example, to convert from French franc to Italian lira, the common currency is defined as European euro. Therefore, in order to convert balances from French franc to Italian lira, balances are converted from French franc to European euro and from European euro to Italian lira.

triggers  An Essbase feature whereby data is monitored according to user-specified criteria which when met cause Essbase to alert the user or system administrator.

tuple  MDX syntax element that references a cell as an intersection of a member from each dimension. If a dimension is omitted, its top member is implied. Examples: (Jan); (Jan, Sales); ([Jan], [Sales], [Cola], [Texas], [Actual])

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.

typed measure  In Essbase, a member tagged as “Text” or “Date” in the dimension where measures are represented. The cell values are displayed as predefined text or dates.

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.

Uniform Resource Locator  The address of a resource on the Internet or an intranet.

unique member name  A non-shared member name that exists only once in a database outline.

unique member outline  A database outline that is not enabled for duplicate member names.

upgrade  The process of replacing an earlier software release with a current release or replacing one product with another.

upper-level block  A type of data block wherein at least one of the sparse members is a parent-level member.

user directory  A centralized location for user and group information. Also known as a repository or provider.

user variable  Dynamically renders data forms based on a user’s member selection, displaying only the specified entity. For example, user variable named Department displays specific departments and employees.

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 member list**  A named, static set of members within a dimension defined by the user.

**validation**  A process of checking a business rule, report script, or partition definition against the outline to make sure that the object being checked is valid. For example, in FDM, validation rules ensure that certain conditions are met after data is loaded from FDM to the target application.

**value dimension**  Used to define input value, translated value, and consolidation detail.

**variance**  Difference between two values (for example, planned and actual value).

**varying attribute**  An attribute association that changes over one or more dimensions. It can be used to track a value in relation to these dimensions; for example, the varying attribute Sales Representative, associated with the Product dimension, can be used to track the value Customer Sales of several different sales representatives in relation to the Time dimension. Varying attributes can also be used for member selection, such as finding the Products that a Sales Representative was responsible for in May.

**version**  Possible outcome used within the context of a scenario of data. For example, Budget - Best Case and Budget - Worst Case where Budget is scenario and Best Case and Worst Case are versions.

**view**  Representation of either a year-to-date or periodic display of data.

**visual cue**  A formatted style, such as a font or a color, that highlights specific types of data values. Data values may be dimension members; parent, child, or shared members; dynamic calculations; members containing a formula; read only data cells; read and write data cells; or linked objects.

**Web server**  Software or hardware hosting intranet or Internet Web pages or Web applications.

**weight**  Value assigned to an item on a scorecard that indicates the relative importance of that item in the calculation of the overall scorecard score. The weighting of all items on a scorecard accumulates to 100%. For example, to recognize the importance of developing new features for a product, the measure for New Features Coded on a developer’s scorecard would be assigned a higher weighting than a measure for Number of Minor Defect Fixes.

**wild card**  Character that represents any single character or group of characters (‘*’) in a search string.

**WITH section**  In MaxL DML, an optional section of the query used for creating re-usable logic to define sets or members. Sets or custom members can be defined once in the WITH section, and then referenced multiple times during a query.

**work flow**  The steps required to process data from start to finish in FDM. The workflow consists of Import (loading data from the GL file), Validate (ensures all members are mapped to a valid account), Export (loads the mapped members to the target application), and Check (verifies accuracy of data by processing data with user-defined validation rules).

**workbook**  An entire spreadsheet file with many worksheets.

**Workspace Page**  A page created with content from multiple sources including documents, URL, and other content types. Enables a user to aggregate content from Oracle and non-Oracle sources.

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

**ws.conf**  A configuration file for Windows platforms.

**wsconf_platform**  A configuration file for UNIX platforms.

**XML**  See Extensible Markup Language (XML).

**XOLAP**  An Essbase multidimensional database that stores only the outline metadata and retrieves all data from a relational database at query time. XOLAP supports aggregate storage databases and applications that contain duplicate member names.

**Y axis scale**  Range of values on Y axis of charts displayed in Investigate Section. For example, use a unique Y axis scale for each chart, the same Y axis scale for all Detail charts, or the same Y axis scale for all charts in the column. Often, using a common Y axis improves your ability to compare charts at a glance.

**Zero Administration**  Software tool that identifies version number of the most up-to-date plug-in on the server.

**zoom**  Sets the magnification of a report. For example, magnify a report to fit whole page, page width, or percentage of magnification based on 100%.
**ZoomChart** Used to view detailed information by enlarging a chart. Enables you to see detailed numeric information on the metric that is displayed in the chart.
## Index

A

abs (SQL function), 295

accounts dimensions
  editing members in, 188
  rules for using, 161
  selecting as dimension type, 163
  specifying in cube schema, 139

adding
  bindings to alias sets, 116
  joins by inspection in minischema, 86
  joins in minischema, 84

Administration Services Console
  configuring, 37
  launching, 221
  prerequisites to launch, 221

aggregate storage
  enabling in Essbase models, 147
  option in members, 186

alias sets
  bindings
    adding, 116
    creating, 114
    deleting, 116
    modifying, 115
  copying, 117
  creating, 114
  deleting, 117
  displaying, 194
  editing, 115
  enabling in Essbase models, 153
  managing, 116
  overview, 113
  renaming, 117
  selecting tables as, 153
  working with, 113

aliases
  accessing properties tabs, 192
  changing and reformatting, 194
  changing cases, 196
  editing members, 192
  Integration Services catalog migration, 255
  multiple, 193
  naming conventions, 272
  overview, 193
  prefixes, 197
  reformatting spaces, 197
  renaming, 194
  search rules, 194, 195
  sets, displaying, 194
  suffixes, 199
  using, 193

alternate hierarchies, 126

applications
  accessing from Essbase Studio, 221
  naming conventions, 271

arranging tables in minischema work area, 93

associating drill-through reports with Essbase models, 245

Attribute Calculations Dimension, 157

attribute calculations member names format
  setting, 157

attribute calculations overview, 158

attributes
  consolidation methods, 154
  defining in Essbase models, 153
  guidelines for using, 154
  overview, 154
  setting attribute calculations member names format, 157
  setting member names format, 156
  types, 154
  user-defined in dimensions, 173

Average member, 158
B
bindings
  adding to alias sets, 116
  creating in alias sets, 114
  delayed in hierarchies, 123
  deleting from alias sets, 116
boolean constant (CPL expression), 280
Boolean ranges, specifying, 158

C
calculation solve order
  dimensions, 167
  members, 183
calendar hierarchies
  fiscal, 129
  Gregorian, 129
  ISO, 132
  limitations, 266
  manufacturing, 132
  retail, 131
case-sensitive names, 272
catalog. See catalog database
catalog database
  configuring, 22
  defined, 22
  limitations, 263
  permissions, 264
  server.properties file, use of, 22
  setting up, 21
catalog migration. See Integration Services catalog migration
catalog.autoinit, 28
catalog.db, 26
catalog.password, 27
catalog.url, 25
catalog.username, 27
classes (CPL expression), 279
color, applying to minischema, 88
column ordering for drill-through reports, 233
columns
  adding joins, 84
  attributes, 146
  filtering results of joins, 86
  hiding, 64
  names in first row
    incremental updates, 69
    modeling text files, 63
text file data sources, 62
  overriding type, 63
  relational source for metadata elements, 58
  renaming, 64
  represented by metadata elements, 52
  sample data, viewing, 75
  selecting type, 64
  viewing in Data Sources Tab, 18
  viewing in Minischemas tab, 18
  viewing properties
    minischemas, 73, 87
    relational sources, 73
    text file source files, 75
    user-defined tables, 73
com.hyperion.cp.level, 33
Common Platform Language. See CPL
configuring
  Administration Services Console, 37
  Essbase Studio Console to locate help, 40
  Essbase Studio server startup file, 39
  server.properties, 23
Connection Wizard
  overview, 49
  workflow, 50
connections. See data source connections
  consolidation of children members
    methods, 182
    overview, 183
  constants (CPL expressions), 280
  contains (SQL function), 290
copying
  alias sets, 117
  metadata elements, 110
Count member, 157
CPL
  expression
    integer constant, 280
  expressions
    binary, use, 283
    boolean constant, 280
    classes, 279
    constant literals, 280
    constants, 280
    data source connections, 278
    databases with three levels, 279
    databases with two levels, 279
definition and use, 277
elements, syntax, 277
flat files (one level), 279
grouping, 281
logical operators, 282
long constant, 280
mathematical operators, 283
multiple operators, 281
number constant, 280
one-level flat files, 279
operands, 277
operations order, 281
operators, 281
operators, logical, 282
operators, mathematical, 283
operators, string, 285
order of operations, 281
readability, 282
string constant, 280
string operators, 285
syntax elements, 277
three-level databases, 279	
two-level databases, 279
unary, use, 283
user-defined tables, 279
overview, 277
SQL functions
    abs, 295
    contains, 290
    date, 285
    dayOfMonth, 285
    exp, 295
    In, 295
    index, 290
    ITrim, 292
    leftStr, 291
    length, 291
    log10, 296
    lower, 291
    month, 286
    monthName, 286
    monthShortName, 286
    numeric, 295
    pow, 296
    quarter, 287
    quarterAsString, 287
    rightStr, 292
    rTrim, 293
    soundex, 293
    sqrt, 297
    string, 290
    substr, 293
    trim, 294
    upper, 294
    weekday, 288
    weekdayName, 288
    weekdayShortName, 288
    year, 289
    yearShort, 289
creating
    alias sets, 114
    bindings in alias sets, 114
    connections to Essbase, 59
    cube schemas, 137
    data source connections
        for Performance Management Architect sources, 59
        for relational sources, 52
        for text file sources, 61
        workflow, 50
drill-through reports, 231
Essbase models
    from existing cube schemas, 142
    in Cube Schema Wizard, 139
metadata elements
    for Performance Management Architect sources in Connection Wizard, 60
    for relational sources in Connection Wizard, 58
    for text file sources in Connection Wizard, 66
    in Connection Wizard, 52
miniscemas, 80
    adding related tables, 81
    adding tables, 81
    metadata elements from objects, 88
    removing tables, 81
    setting general properties, 80
    specifying filters, 81
cube deployment
    connecting to Essbase Server, 214
    connection and load information, 215
deleting
    databases, 217
    members, 216
    Essbase Server connection, 214
    history, 219
incremental load
  selecting dimensions and members, 217
  setting up, 218
limitations, 267
load information, 215
load options, 213
members, deleting, 216
model properties
  modifying, 217
  providing information, 215
overview, 213
rules and rule files, 217
server information, 215
viewing history, 219
Wizard, 214
cube Linkage, updating, 222
cube schemas
  creating, 137
  creating an Essbase model in Cube Schema Wizard, 139
  creating Essbase models from existing, 142
cube linkage, 222
data load bindings, defining, 140
data load mappings, defining, 140
deployment history, viewing, 142
editing, 137
Essbase models, 146
hierarchies
  previewing, 139
  specifying, 138
measures, specifying, 138
overview, 137
setting options, 139
specifying the accounts dimension, 139
custom data load settings
  in model properties, 148
SQL limitations, 267
customizing
  drill-through reports, 233
  drill-through reports for relational sources, 233
cycle dependency limitation, 265

data load bindings in cube schemas, 140
data load mappings in cube schemas, 140
data load scaling, 191
data load SQL
  editing guidelines, 149
  overriding, 148
  overview, 149
data source connections
  CPL expressions, 278
creating
  for Performance Management Architect sources, 59
  for relational sources, 52
  for text file sources, 61
metadata elements, 52
metadata elements for Performance Management Architect sources, 60
metadata elements for relational sources, 58
metadata elements for text file sources, 66
minischema for relational sources, 56
user-defined tables, 77
workflow described, 50
defining parameters, 50
  for Essbase, 59
  for Performance Management Architect sources, 60
  for relational sources, 53
  for text file sources, 61
deleting, 76
editing connection properties, 71
friendly names, 76
incremental update, 67
  of relational data source, 67
  of text file data source, 69
introspection, 70
minischema options, 51
  for relational sources, 55
  for text file sources, 65
modeling text file sources, 63
overview, 49
populating a minischema
  for relational sources, 57
  for text file sources, 65
refreshing the list, 76
reusing
  minischema for relational sources, 56
  minischema for text file sources, 65
selecting
  minischema option for text file sources, 64
subset of tables, 51
tables for relational sources, 54
skipping a minischema
  for relational sources, 57
  for text file sources, 65
user-defined tables
  example, 78
  overview, 77
viewing
  properties of columns, 73
  properties of source tables and columns, 73
  properties of tables, 73
  sample data, 75
  statement of user-defined table, 74
  text file source columns, 75
  text file source files, 74
working with, 67
data sources
  limitations, 263
  permissions, 264
Data Sources tab, 18
data storage method
  dimensions, 166
  members, 185
databases
  naming conventions, 271
  three level (CPL expressions), 279
  two level (CPL expressions), 279
Date (SQL function), 285
date elements, creating, 109
date ranges, specifying, 158
day attributes in hierarchies, 134
dayOfMonth (SQL function), 285
defining
  connection parameters for Essbase, 59
  data load bindings in cube schemas, 140
  data load mappings in cube schemas, 140
  data source connection parameters
    Connection Wizard, 50
    for Performance Management Architect sources, 60
    for relational sources, 53
    for text file sources, 61
drill-through report type, 233
drill-through report type for relational sources, 233
  Template SQL for drill-through reports, 235
delayed key binding columns in hierarchies, 123
deleting
  alias sets, 117
  bindings from alias sets, 116
  data source connections, 18, 76
  joins in minischemas, 85
  minischemas, 94
  tables from minischemas, 93
dense dimensions, 165
deployed applications, 221
derived text measures
  creating, 100, 107
  editing, 100, 107
  limitations, 265
  naming conventions, 272
dimension elements
  creating
    from Data Sources tab, 101
    from Metadata Navigator, 100
    from Minischemas tab, 101
defining general properties, 102
editing
  general properties, 102
  filters, 105
  key binding, 103
  naming conventions, 272
  overview, 99
  sort order, 106
  viewing sample data for, 75
dimensions
  accounts, 163
  accounts, editing properties, 168
  aliases, 172
  calculation solve order, 167
  data loads, optimizing, 175
  data storage method, 166
definition, 146
dense, 165
duplicate member settings, 175
editing, 163
  facts, 146
  formulas, adding, 171
  hierarchy storage settings, 174
  naming conventions, 272
  outline build options, 174
  overview, 160
  properties, 159
  skip option, 170
  sparse, 165
standard, 163
storage method, 165
tabs on properties dialog box, 160
tabs, accessing, 161
time, 163
time balance, 169
type, 160, 163
user-defined attributes, 173
user-defined attributes, assigning, 173
using accounts dimensions, 161
using time dimensions, 161
variance reporting method, 170
Drill-through Report Properties dialog box
  Associations tab, 245
  Main tab, 231
  Report Contents tab, 233
drill-through reports
  associating with Essbase models, 245
column ordering, 233
creating, 231
customizing, 233
customizing for relational source, 233
defined, 230
defining the report type, 233
Drill-through Report Properties dialog box
  Associations tab, 245
  Main tab, 231
  Report Contents tab, 233
FDM URL target, 239, 240
intersection levels, specifying, 231
Java Method type, defining, 243
limitations, 268, 276
OBI URL target, 239, 242
report type, defining for relational source, 233
row governors, defining for relational sources, 237
Sample FDM URL Template, 239, 240
Sample OBI URL Template, 239, 242
Sample URL Template, 239, 240
specifying filters for relational sources, 237
Template SQL, defining, 235
testing for relational sources, 238
URL target, 239, 240
URL type, defining, 238
duplicate member names, enabling, 148
duplicate member settings, moving, 175
dynamic calc
  attribute member names, 157
data storage in dimensions, 167
  members, 185
dynamic time series, 164

E
EASLaunch.properties, 37
editing
  accounts, dimension properties, 168
  alias sets, 115
  bindings in alias sets, 115
cube schemas, 137
data source connection properties, 71
dimensions, 163
  hierarchies, 135
  members, 182
  minischemas
    adding related tables, 81
    adding tables, 81
    joins, 84
    properties, 93
    removing tables, 81
    specifying filters, 81
Essbase Administration Services
  see Administration Services, 221
Essbase applications, accessing from Essbase Studio, 221
Essbase model properties
  accessing dialog box, 147
  limitations, 267
  setting, 147
tabs in dialog box, 145
Essbase models
  account dimension properties, 168
  alias properties, setting, 153
  associating drill-through reports, 245
  attributes
    calculation member name format, 157
    defining, 153
    member names format, 156
  Boolean ranges, 158
  browsing, 210
  changes, reviewing, 210
  characteristics of, 146
columns
  attributes, 146
tagging, 146
components, 146
creating from existing cube schemas, 142
custom data load settings, 148
Custom SQL, using, 149
data loads, optimizing, 175
date ranges, 158
definition, 146
dimensions
  aliases, 172
  properties, 159
  storage method, 165
type, 163
  use in models, 146
duplicate member names, enabling, 148
duplicate member settings, moving, 175
dynamic time series, 164
  enabling for XOLAP, 152
formulas
  adding to dimensions, 171
  adding to members, 192
hierarchy storage settings, 174
members
  aliases, 192
  properties, 176
numeric ranges, 158
opening work area, 208
optimizing data loads, 175
outline build options, 174
overview, 146
reusing, 146
reviewing changes, 210
saving, 148
setting properties, 147
skip option
  in dimensions, 170
  in members, 189
tabs on dialog box, 146
time balance
  in dimensions, 169
  in members, 189
tool tips, 211
two pass calculation, 165
Unicode, enabling, 148
user-defined attributes, 173
using work area, 208
validating properties, 209
variance reporting method
  in dimensions, 170
  in members, 190
viewing, 207
work area, 208
working with, 207
XOLAP, enabling, 148
Essbase Properties dialog box, 146
Essbase Studio
  administration overview, 21
catalog database, 21
  compatibility with other products, 21
drill-through, 17
  Essbase compatibility, 21
help, viewing, 40
introduction, 17
lineage tracking, 17
logging in, 20
metadata catalog, 17
Metadata Navigator, 19
  overview, 17
server. See Essbase Studio server
setting up catalog database, 21
Smart View compatibility, 21
Source Navigator, 18
starting server and console, 40
stopping server and console, 40
user interface, 17
Essbase Studio Console
  online help files, locating, 40
reconnecting to Essbase Studio Server, 20
starting, 40
stopping, 40
Essbase Studio Server
  access limitations, 263
  commands
    dumps, 41
    exit, 43
    overview, 41
    pconf, 42
    squeue, 42
    sres, 43
    sthd, 43
    version, 41
  limitations to accessing, 263
reconnecting from Essbase Studio Console, 20
starting, 40
stopping, 40
Essbase Studio server startup file, configuring, 39
Essbase, creating connections to, 59
example minischema, 80
Excel files as data source
limitations, 265
exp (SQL function), 295
F
facts
in models, 146
tables, 146
FDM URL drill-through report target, 239, 240
filters
dimension elements, 105
introspection
dimension table, 97
fact table, 97
specifying in drill-through reports for relational sources, 237
find and search
difference in functions, 247
overview, 247
finding, 247. See also searching
applying to current selection, 248
matching case, 248
matching word, 248
metadata elements, 247
metadata elements, overview, 247
fiscal calendar hierarchies, 129
flat file (CPL expression), 279
flat file sources
see text file sources, 61
formulas
adding to dimensions, 171
adding to members, 192
overview, 172
friendly names of data source connections, showing, 76
G
general preferences, setting, 251
Gregorian calendar hierarchies, 129
grouping (CPL expression)
aiding readability, 282
connections, 282
examples, 282
to associate subexpressions, 281
H
hierarchies
alternate, 126
alternate in outline builds, 174
calendar
creating, 128
fiscal, 129
Gregorian, 129
ISO, 132
manufacturing, 132
retail, 131
children and siblings, adding, 122
cube deployment, using in, 218
day attributes, 134
determining structure of Essbase models, 119
editing, 135
elements, adding, 121
examples
alternate, 126
multi-chain, 126
multi-chain with attribute dimensions, 126
recursive, 128
single-chain, 125
single-chain, multi-generation, 125
time built from DATE element, 128
fiscal calendar, creating, 129
Gregorian calendar, creating, 129
ISO calendar, creating, 132
limitations, 266
linked value attributes, 135
manufacturing calendar, creating, 132
measure, creating, 120
metadata elements, adding, 121
multi-chain, 126
multi-chain with attribute dimensions, 126
overview, 119
physical elements, adding, 121
previewing in cube schemas, 139
recursive, 128
retail calendar, creating, 131
single-chain, 125
single-chain, multi-generation, 125
specifying for cube schemas, 138, 146
standard, creating, 120
storage settings, selecting, 174
time built from DATE element, 128
types, 125
I
In (SQL function), 295
incremental update
  existing data source, 67
  existing relational data source, 67
  existing text file data source, 69
independent dimension bindings
  defining, 180
  limitations, 267
index (SQL function), 290
integer constant (CPL expression), 280
Integration Services catalog migration
  accounts dimension, 255
  aliases, mapping, 255
  dialog box, accessing, 255
  dimensions, 254
  Essbase export model, 255
  hierarchies, mapping, 254
  Integration Services Models, 255
  limitations and restrictions, 257
  measure column, mapping, 254
  metadata, 253
  metaoutlines, 254, 255
  models, 253
  overview, 253
  procedure, 256
  properties not migrated, 258
  reports, mapping, 255
  user-defined members, mapping, 255
intersection levels
  specifying in drill-through reports, 231
introspection
  limitations, 265
  overview, 95
  performing on existing data source, 70
  selecting
    dimension table, 97
    fact table, 96
    hierarchies, 97
    minischema option, 96
ISO calendar hierarchies, 132
J
Java method, drill-through report type, 243
joins
  adding by inspection in minischema, 86
  adding in minischema, 84
  editing in minischema, 84
L
label only option, 186
last deployed field, 147
laying out tables in minischema work area, 93
leftStr (SQL function), 291
length (SQL function), 291
limitations
  accessing Essbase Studio Server, 263
  adding physical table to minischema, 87
  calendar hierarchies, 266
  catalog, 263
  cube deployment, 267
  cycle dependency, 265
  data sources, 263
  drill-through reports, 268, 276
  Excel files as data source, 265
  for derived text measures, 265
  for XOLAP models, 265
  hierarchies, 266
  introspection, 265
  measure hierarchies, 266
  metadata elements, 265
  MySQL, 265
  standard hierarchies, 266
lineage
  elements displayed in, 225
  overview, 225
  work area
    displaying elements, 226
    navigating, 226
    opening, 226
    using, 226
linked value attributes in hierarchies, 135
log10 (SQL function), 296
logger.count, 30
logger.file, 29
logger.limit, 30
logging in to Essbase Studio, 20
long constant (CPL expression), 280
lower (SQL function), 291

Index 331
M

manufacturing calendar hierarchies, 132
maximizing the minischema work area, 92
Maximum member, 157
measure hierarchies, 120
measures
  specifying for cube schemas, 138
members
  accounts, editing properties, 188
  aggregate storage option, 186
  aggregate storage with label only option, 186
  alias sets, displaying, 194
  aliases editing, 192
  attribute settings, 178
  average, 158
  calculation solve order, 183
  changing cases, 206
  consolidation methods for children, 182
  data storage method, 185
  editing, 182
  formulas, adding, 192
  multiple aliases, 193
  names, changing and reformatting, 204
  naming conventions, 272
  naming sequence, 201
  overview, 176
  prefixes, 201
  properties, 176
  reformatting, 207
  renaming, 204
  renaming sequence, 201
  search rules, 204, 205
  setting attribute calculations member names format, 157
  setting general properties, 177
  setting names format, 156
  skip option, 189
  suffixes, 202
  tabs, accessing, 176
  time balance, 189
  transforming, 204
  two pass calculations, 185
  user-defined attributes, 200
  variance reporting method, 190
  varying attributes, 178
metadata catalog, 17
metadata elements
  copying, 110
  creating, 100
    for Performance Management Architect sources, 60
    for relational sources, 58
    for text file sources, 66
    from minischema objects, 88
    in Connection Wizard, 52
  deleting, 111
  editing, 100
  finding, 247
  finding, overview, 247
  hierarchies, adding in, 121
  limitations, 265
  lineage, showing, 111, 225
  overview, 99
  renaming, 111
  sample data for, 111
  searching, 248
  searching, overview, 247
  types, 99
  working with, 110
metadata folders
  creating, 110
  editing, 110
Metadata Navigator
  display of elements, 19
  overview, 19
  source of elements, 19
  types of elements created, 20
migration. See Integration Services catalog migration
minimizing the minischema work area, 92
Minimum member, 157
Minischema Work Area. See Work Area
minischemas, 79
  adding joins, 84
  adding joins by inspection, 86
  adding tables to, 86
  arranging tables in minischema work area, 93
  color, applying, 88
  columns, viewing properties, 73
  creating, 80
    adding related tables, 81
    adding tables, 81
    filters, specifying, 81
    removing tables, 81
  setting general properties for, 80
creating for relational sources in Connection Wizard, 56
deleting, 94
editing
  adding related tables, 81
  adding tables, 81
  filters, specifying, 81
  removing tables, 81
editing joins, 84
editing properties, 93
example, 80
laying out tables in minischema work area, 93
maximizing the work area, 92
metadata elements, creating from minischema objects, 88
minimizing the work area, 92
modeling text file sources in Connection Wizard, 63
opening, 83
options
  for relational sources in Connection Wizard, 55
  for text file sources in Connection Wizard, 65
overview, 79
populating
  for relational sources in Connection Wizard, 57
  for text file sources in Connection Wizard, 65
  in Connection Wizard, 51
refreshing the list, 94
removing tables from, 83
reusing for relational sources in Connection Wizard, 56
reusing for text file sources in Connection Wizard, 65
saving, 83
selecting option for text file sources, 64
selecting options in Connection Wizard, 51
skipping
  for relational sources in Connection Wizard, 57
  for text file sources in Connection Wizard, 65
tables, viewing properties of, 73
viewing, 89
viewing sample data in, 87
work Area, using, 89
working with, 82
zooming in, 93
zooming out, 93
Minischemas tab, 18
modeling text file data sources, 63
modifying. See editing
month (SQL function), 286
monthName (SQL function), 286
monthShortName (SQL function), 286
multi-chain hierarchies, 126
multi-chain hierarchies with attribute dimensions, 126
multi-chain hierarchies with shared members. See alternate hierarchies
MySQL, limitations, 265

N
names in first row of columns
  incremental updates, 69
  modeling text files, 63
text file data source, 62
naming conventions
  aliases, 272
  applications, 271
databases, 271
derived text measures, 272
dimension elements, 272
dimensions, 272
drill-through reports, 276
members, 272
reserved words, 274
restricted characters, 273
Navigator
  Metadata, 19
  Source, 18
number constant (CPL expression), 280
numeric (SQL function), 295
numeric ranges, specifying, 158

O
OBI URL drill-through report target, 239, 242
OLAP
  overview, 150
  use with multidimensional databases, 150
opening
  minischemas, 83
operands (CPL expressions), 277
operators
  CPL expressions, 281
  function, 281
grouping, 281
logical, 282
mathematical, 283
multiple, 281
order, 281
sequence, 281
string, 285
optimizing data loads, 175
outline build options, 174

P
parent-child hierarchies. See recursive hierarchies
Performance Management Architect sources
  creating connections to, 59
  creating metadata elements for, 60
  defining connection parameters for, 60
permissions
  catalog database, 264
  data sources, 264
pow (SQL function), 296
preferences
  setting general, 251
  setting schema, 252
prefixes
  aliases, adding, 197
  members, adding, 201
previewing
  sample data for dimension elements, 75
  sample data in data sources, 75
  sample data in minischema, 87
properties
  editing data source connection, 71
  editing in minischema, 93
  setting during minischema creation, 80
  setting Essbase models, 147

Q
quarter (SQL function), 287
quarterAsString (SQL function), 287

R
reconnecting to Essbase Studio Server, 20
recursive hierarchies, 128
refreshing
  data source connection list, 76
  minischema list, 94
relational source columns, viewing properties, 73
relational source tables, viewing properties, 73
relational sources
  creating
    connections to, 52
    metadata elements, 58
    minischema, 56
  defining connection parameters for, 53
  populating a minischema for, 57
  reusing an existing minischema for, 56
  selecting a minischema option, 55
  skipping minischema creation, 57
renaming
  alias sets, 117
  members, 204
  transformation sequence, 201
reserved words in names, 274
restricted characters in names, 273
retail calendar hierarchies, 131
rightStr (SQL function), 292
row governors, defining for relational sources, 237
r’Tim (SQL function), 293

S
sample data
  viewing for dimension elements, 75
  viewing in data sources, 75
  viewing in minischema, 87
Sample FDM URL Template for drill-through reports, 239, 240
Sample OBI URL Template for drill-through reports, 239, 242
Sample URL Template for drill-through reports, 239, 240
saving
  dimension
    properties, 162
    storage method, 165
    type, 164
duplicate member settings, 175
Essbase models, 148
formulas in dimensions, 172
hierarchy storage settings, 174
minischemas, 83
server.properties file, 24
skip option, 171
time balance, 169
user-defined attributes in dimensions, 173
variance reporting method, 170
scaling, data load, 191
schema. See minischemas
schema preferences, setting, 252
scraping. See introspection.
search rules
  aliases, 194, 195
  members, 204, 205
searching, 248. See also finding
  for metadata elements, 248
  Look In option, 249
metadata elements, overview, 247
type, 248
server properties file. See server.properties
server.charset, 32
server.css.URL, 26
server.datafile.dir, 28
server.essbase.disableDistinct, 34
server.essbase.streamingCubeBuilding, 28
server.essbase.TPTapi, 34
server.hss.bpmApplication, 27
server.queueSize, 30
server.resourceCount, 31
server.sql.fetchSize, 32
server.tempDir, 32
server.threadCount, 31
server.timeoutPeriod, 29
server.writeLockTimeOut, 33
server.resourceCount, 31
server.sql.fetchSize, 32
server.tempDir, 32
server.threadCount, 31
server.timeoutPeriod, 29
server.writeLockTimeOut, 33
setting
cube schema options, 139
general preferences, 251
schema preferences, 252
single-chain hierarchies, 125
skip option
  in dimensions, 170
  in members, 189
solve order, members, 184
soundex (SQL function), 293
Source Navigator
  Data Sources tab, 18
  Minischemas tab, 18
  tabs, 18
  work area, 19
source tables and columns, viewing properties, 73
sparse dimensions, 165
specifying
column ordering for drill-through reports, 233
drill-through report filters for relational sources, 237
hierarchies for cube shemas, 138
intersection levels in drill-through reports, 231
measures for cube shemas, 138
member properties in Essbase models, 177
row governors for relational sources, 237
SQL functions. See CPL SQL functions


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

sqrt (SQL function), 297
standard dimensions, 163
standard hierarchies, 120
startServer.bat, 39
string (SQL function), 290
string constant (CPL expression), 280
Studio. See Essbase Studio
substr (SQL function), 293

suffixes
   aliases, adding, 199
   members, adding, 202
Sum member, 157
syntax elements (CPL expressions), 277

T
tables
   adding a subset in Connection Wizard, 51
   adding joins, 84
   adding related to minischema, 93
   adding to minischema, 86
   applying color in minischemas, 88
   arranging in minischema, 93
   creating metadata elements from, 88
   filtering results of joins, 86
   removing from minischema, 83, 93
   sample data, 87
   selecting relational tables in Connection Wizard, 54
   user-defined, viewing properties, 73
   user-defined, viewing statements, 74
   viewing sample data in, 87
Template SQL, defining for drill-through reports, 235
testing
   drill-through reports for relational sources, 238
text file source columns, viewing properties, 75
text file source files, viewing properties, 74
text file sources
   columns, viewing properties of, 75
   creating connections to, 61
   creating metadata elements for, 66
   defining connection parameters for, 61
   modeling, 63
   populating a minischema for, 65
   reusing an existing minischema for, 65
   selecting a minischema option, 65
   skipping minischema creation, 65
   source files, viewing properties of, 74
three level database (CPL expression), 279
time balance
   in dimensions, 169
   in members, 189
   overview, 169
time dimensions, 161, 163
time hierarchies built from DATE element, 128
tool tips, 211
transformation of members, 204
trim (SQL function), 294
two level database (CPL expression), 279
two pass calculation
   in dimensions, 165
   in members, 185
   overview, 166
   selecting for dimensions, 165

U
Unicode, in Essbase models, 148
updating cube linkage, 222
upper (SQL function), 294
URL drill-through report target, 239, 240
URL drill-through report type, 238
user-defined attributes
   assigning to dimensions, 173
   assigning to members, 200
   guidelines in using, 173
   overview, 173
user-defined source columns, viewing properties, 73
user-defined tables
   CPL expressions, 279
   creating, 77
   example, 78
   overview, 77
   viewing properties, 73
   viewing source data, 77
   viewing table statements, 74

V
variance reporting method
   in dimensions, 170
   in members, 190
varying attributes
   example scenario, 155
   independent dimension bindings, defining, 180
independent dimension bindings, limitations, 267
overview, 155
setting for members, 178
viewing
cube deployment history, 219
cube schema deployment history, 142
Essbase Studio help, 40
maximizing the minischema work area, 92
minimizing the minischema work area, 92
minischemas
work area, 89
zooming in, 93
zooming out, 93
properties of source tables, 73
sample data for dimension elements, 75
sample data in data sources, 75
sample data in minischema, 87

W
weekday (SQL function), 288
weekdayName (SQL function), 288
weekdayShortName (SQL function), 288
work area, 19
objects displayed in, 19
opening, 208
using, 208
using in minischemas, 89

X
XOLAP
designating a model for, 152
enabling in Essbase models, 148
guidelines for using, 151
model limitations, 265
overview, 151
workflow, 151

Y
year (SQL function), 289
yearShort (SQL function), 289

Z
zooming in on minischemas, 93
zooming out on minischemas, 93