Oracle® Essbase Studio

User's Guide

RELEASE 11.1.2.2.000

ORACLE®

ENTERPRISE PERFORMANCE
MANAGEMENT SYSTEM

Essbase Studio User's Guide, 11.1.2.2.000

Authors: EPM Information Development Team

Contents

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

1 Essbase Studio Overview

In This Chapter

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:

- **Data Source Navigator**, described in "Data Source Navigator" on page 20
- Work area, described in "Work Area" on page 21

- **Metadata Navigator**, described in "**Metadata Navigator**" on page 21

Additionally, you can choose to display or dock Console Messages.

Data Source Navigator

The **Data Source Navigator**, displayed by default in the right pane of the Essbase Studio Console lists the physical data sources to which you have created connections. This structure is often referred to as the *physical tree*. You can launch the **Connection Wizard** from here, where you create data source connections. For each data source connection, the associated minischemas are also displayed. Minischemas are 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.

Other tasks that you can perform on data sources are:

- 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
- Create, edit, and manage minischemas.

  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.

  - 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 on data source connections and minischemas, see:

- Chapter 4, "Data Source Connections"
- 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 "About Minischemas" on page 111
- Essbase models, described in Chapter 11, "Essbase Properties"
- Lineage View, described in Chapter 15, "Lineage"
- Drill-through reports, described in Chapter 16, "Drill-through Reports"
- Hierarchy editor, described in Chapter 9, "Hierarchies"
- Sample data, described in the topic, "Viewing Sample Data" on page 104 in Chapter 4, "Data Source Connections"
- Deployment history, described in "Viewing Deployment History" on page 275

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 files, Oracle BI EE, 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
- Text lists
- 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.**
3. **Optional:** Select **Remember this user.**

   On subsequent logins, you can select your user name from the drop-down list.

4. Click **Log In**

Note that after logging in, the title bar of the Essbase Studio Console window displays this information:

- User name
- Server name
- User role

You are now ready to begin working 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**, and then **Reconnect To Server** to display the **Login** dialog box.

The **Server** and **User** fields are read only.

2 Enter your **Password**, then click **Log In.**

2 Administration

In This Chapter

Administration Overview

After you install Essbase Studio using the Oracle Hyperion Enterprise Performance Management System Installer, Fusion Edition, you configure using the Oracle's Hyperion Enterprise Performance Management System Configurator. Oracle recommends that, where possible, you always use the EPM System Configurator whenever you need make changes your configuration. You may, however, manually edit some of the Essbase Studio configuration files.

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 Oracle Hyperion Enterprise Performance Management System 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 EPM System 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 *`MIDDLEWARE_HOME`*`/user_projects/epmsystem1/BPMS/bpms1/bin` 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 used for your Essbase Studio catalog database, Oracle recommends setting the following privileges for the user that is used to connect to catalog database.

  ```
  open_cursors=nnn SCOPE=MEMORY
  ```

  where `nnn` >= 300.

If you are using one database instance for all EPM System 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 by 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** *MIDDLEWARE_HOME*/user_projects/epmsystem1/BPMS/bpms1/bin**, 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 33.
- catalog.password—The encrypted password for the catalog database user. See "catalog.password" on page 33.

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 28 for information on configuration options.

Upgrading the Essbase Studio Catalog and Data

To move Essbase Studio from release 11.1.1.3 and earlier to the 11.1.2.1 release, you perform an upgrade. To move from release 11.1.2 to 11.1.2.1, you apply the maintenance release.

Before you begin the upgrade or maintenance release procedures for Essbase Studio, complete the "Upgrading Checklist" tasks in the *Oracle Hyperion Enterprise Performance Management System Installation and Configuration Guide*.

Note: You must complete the upgrade tasks applicable to your environment in the order noted in the checklist.

After you complete the applicable EPM System environment upgrade tasks, you may begin the Essbase Studio upgrade or applying the maintenance release.

Whether you are upgrading or applying the maintenance release, you must migrate your catalog database to the current version to keep information in the catalog intact. Information such as data source connections, metadata elements, cube schemas, and Essbase models will be functional and deployable after the migration and subsequent upgrade tasks.

Perform the Essbase Studio catalog database upgrade during the EPM System configuration process. The instructions for migrating the Essbase Studio catalog database are included in the *Oracle Hyperion Enterprise Performance Management System Installation and Configuration Guide*.

For upgrades (from release 11.1.1.3 and earlier), after configuration, perform these tasks:

- If Essbase Studio Server has moved, for deployed applications that reference the old server location, update the cube linkage, as described in "Updating Cube Linkage" on page 276.
- If Essbase Server has moved, perform the rehosting procedure so that Essbase connections point to the new server location, as described in "Updating References to Rehosted EPM System Products" on page 62.
- For text file data sources:
  - Oracle recommends that you store the text files in the default directory (*EPM_ORACLE_HOME*`/products/Essbase/EssbaseStudio/Server/datafiles`) on the machine where Essbase Studio Server resides. If you choose to store text files in a non-default location, you must add the `server.datafile.dir` property to the `server.properties` file, as described in "server.datafile.dir" on page 34.
  - Also, if you are storing text files in a non-default directory, edit the connection properties for any text file connections to point to the new location of the text files, as described in "Editing Data Source Connection Properties" on page 99.
- If Performance Management Architect Dimension Server has moved, edit the connection properties for any Performance Management Architect Dimension Server connections that point to the old server location, as described in "Editing Data Source Connection Properties" on page 99.

Enabling SSL for Essbase Studio

Essbase can be deployed to work in Secure Socket Layer (SSL) and non-SSL modes. In SSL mode, all communication between Essbase Server and Essbase Studio Server is encrypted to ensure data security. Default deployments of Essbase components install self-signed certificates to enable SSL communication, mainly for testing purposes. Oracle recommends that you use certificates from well-known third-party certification authorities (CAs) to SSL-enable Essbase in production environments. See the *Oracle Hyperion Enterprise Performance Management System Security Administration Guide* for more information.

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 edit or add server properties.

Note: The `server.properties` file is located in *MIDDLEWARE_HOME*/user_projects/epmsystem1/BPMS/bpms1/bin.

For information on *MIDDLEWARE_HOME*, see the *Oracle Hyperion Enterprise Performance Management System Installation and Configuration Guide*.

➤ 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 do not need to 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 30 for descriptions and examples of each property.

```
catalog.url=database tag\://host:port/
server.css.URL=server.css.URL=http\://server.us.oracle.com\:port/interop/framework/
getCSSConfigFile
catalog.db=catalog database name
catalog.username=catalog user ID
catalog.password=encrypted catalog password
server.hss.bpmApplication=BPM\:BPM application ID
server.datafile.dir=path to flat file directory
server.essbase.streamingCubeBuilding=true|false
server.timeoutPeriod=number of seconds
server.queueSize=size of the task queue
server.threadCount=number of worker threadsserver.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
server.essbase.TPTapi=true|false
server.essbase.disableDistinct=true|false
server.httpPort=port number
transport.port=port number
server.runInBackground=true|false
server.essbase.blindShare=true|false
oracle.jdbc.ReadTimeout=milliseconds
data-source-type.pool.maxsize=maximum size of the connection pool
data-source-type.cache.size=cache size
server.essbase.uniqueMemberFromCaptionBinding=true|false
```

Note: Some exceptions to this syntax exist, depending on your RDBMS. They are discussed in "Server Properties File Examples" on page 44.

Server Properties

Following are the configurable server properties:

catalog.url

The URL for the catalog database.

Syntax

```
catalog.url=database_tag\://host:port/
```

where *host* is a server name or server plus domain name, depending on your requirements.

Notes

- The following database tags are supported:
  - `oracle`—Oracle
  - `db2`—IBM DB2
  - `sqlserver`—Microsoft SQL Server
- When your catalog database is Oracle using an Oracle ID (SID), the Oracle SID is appended to the `catalog.url` parameter, as follows:

  ```
  catalog.url=oracle\://host\:port\:OracleSID
  ```
- When your catalog database is Oracle using an Oracle Service Name, the Oracle Service Name is appended to the `catalog.url` parameter, as follows:

  ```
  catalog.url=oracle\://host\:port/OracleServiceName
  ```
- When your catalog database is located in an Oracle RAC instance, the `catalog.url` parameter appears as follows:

  ```
  catalog.url=
  oracle\:// (DESCRIPTION=(LOAD_BALANCE=on)
  (ADDRESS=(PROTOCOL=TCP)(HOST=host1)(PORT=port-of-host1
  [ (ADDRESS=(PROTOCOL=TCP)(HOST=host2)(PORT=port-of-host2))
  [ (ADDRESS=(PROTOCOL=TCP)(HOST=hostN)(PORT=port-of-hostN)) ]...]
  (CONNECT_DATA=(SERVICE_NAME=service)))
  ```
- When your catalog database is IBM DB2, the database name is appended to the `catalog.url` parameter, using the following syntax:

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

Examples

- Oracle using SID:

  ```
  catalog.url=oracle\://sequoia.efgco.com\:1521\:bpm
  ```

  ```
  catalog.url=oracle\://sequoia\:1521\:bpm
  ```
- Oracle using Oracle Service Name:

  ```
  oracle://aspen123.us.xyzco.com\:1521/epmdb.us.xyzco.com
  ```

  ```
  oracle://aspen123\:1521/epmdb.us.xyzco.com
  ```
- Oracle RAC:

  ```
  catalog.url=oracle\://(DESCRIPTION=(LOAD_BALANCE=on)(ADDRESS=(PROTOCOL=TCP)
  (HOST=host1)(PORT=1521))(ADDRESS=(PROTOCOL=TCP)(HOST=host2)(PORT=1521))
  (CONNECT_DATA=(SERVICE_NAME=service)))
  ```
- IBM DB2:

  ```
  catalog.url=db2\://cypress.abcco.com\:50000;databaseName=EPM_Db
  ```

```
catalog.url=db2\://cypress\:50000;databaseName=EPM_Db
```

- Microsoft SQL Server:

```
catalog.url=sqlserver\://aspen3.us.oracle.com\:1433
```

```
catalog.url=sqlserver\://aspen3\:1433
```

See “Server Properties File Examples” on page 44 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

```
server.css.URL=http\://aspen3.us.oracle.com\:28080/interop/framework/getCSSConfigFile
```

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 EPM 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
```

server.datafile.dir

Defines the root directory for flat files that will serve as a data source for Essbase Studio Server. The default storage location for text files is:

EPM_ORACLE_INSTANCE/BPMS/bpms1/datafiles

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

Syntax

`server.datafile.dir=`*`path to flat file directory`*

Examples

- **Windows:**

  Text files stored on the local Essbase Studio Server machine:

  ```
  server.datafile.dir=C:\\EssbaseStudio\\text_file_sources
  ```

  Text files stored on a UNC (Universal Naming Convention) path:

  ```
  server.datafile.dir=\\\\svr33\\cedar5\\EssbaseStudio\\text_file_sources
  ```

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

- **UNIX:**

  ```
  server.datafile.dir=/vol1/cedar5/EssbaseStudio/text_file_sources
  ```

Notes

- Text file data source directory paths are limited in length as follows:
  - Native mode: 121 bytes
  - Unicode mode: 1028 bytes
- If `server.datafile.dir` is changed, then Essbase Studio Server must be restarted for the modification to take effect.

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.

When `server.essbase.streamingCubeBuilding` is set to false, streaming mode can be selected at deployment time in the Cube Deployment Wizard (described in step 5 in "Providing Connection Information for Cube Deployment" on page 266).

IBM DB2 users: In the "Define connections" page of the Connection Wizard, when you specify an authentication method other than "No Encryption" for a data source, the `server.essbase.streamingCubeBuilding` property must be set to "true" to enable deployments in streaming mode. See "Defining Connection Parameters for Relational Sources" on page 75 for more information on specifying an authentication method.

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
```

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 36.

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*

If a "name too long" error is returned when attempting to deploy a cube, add an entry for the `server.tempDir` property to the `server.properties` file, and specify a shorter directory path. For example:

```
server.tempDir=C:\\studiotemp
```

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 38.

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 37.

Syntax

server.writeLockTimeOut=*number of seconds*

Notes

See the Notes section in "server.readLockTimeOut" on page 37.

Example

```
server.writeLockTimeOut=150
```

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

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

Example

`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

`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.
- **IBM DB2 users:** A limitation in IBM DB2's handling of the LONG VARCHAR data type in a select DISTINCT statement causes cube deployment to fail. To avoid this, set `server.essbase.disableDistinct` to true.

Example

`server.essbase.disableDistinct=true`

server.httpPort

The HTTP port on which Essbase Studio Server listens. The HTTP port is used by Essbase Studio Server to communicate with Oracle Essbase Spreadsheet Add-in during drill-through operations, and to communicate with Performance Management Architect.

The default HTTP port number is 9080.

If you change the default setting of the `server.httpPort` property, you will not be able to perform drill-through operations in Spreadsheet Add-in. Drill-through in Oracle Hyperion Smart View for Office, Fusion Edition is not affected.

If there is a port conflict with other applications or programs running on the Essbase Studio Server computer, it is recommended that you change the port number of the conflicting application.

However, if you must change the `server.httpPort` setting, you may do so by adding an entry for the `server.httpPort` property to the `server.properties` file.

Syntax

`server.httpPort=`*`port number`*

Example

```
server.httpPort=9080
```

transport.port

Specifies the TCP port on which Essbase Studio Server listens.

The default port number is 5300.

If there is a port conflict with other applications or programs running on the Essbase Studio Server computer, you may change the Essbase Studio Server port number by adding an entry for the `transport.port` property to the `server.properties` file.

Syntax

transport.port=*port number*

Example

In this example, the port number is changed to port 3000.

```
transport.port=3000
```

Notes

When port conflicts arise, it is recommended that you run the EPM System Configurator to change the port assignment for Essbase Studio Server.

If you change the `transport.port` property manually, without running the EPM System Configurator, you must enter the port number for Essbase Studio Server when logging in to the console.

For example, for an instance of Essbase Studio Server running on a machine named "aspen," if you changed the port assignment from the default to port number 1234, you would enter the following in the Server field of the Login dialog box when logging in to the Essbase Studio Console:

```
aspen:1234
```

server.runInBackground

When set to "true," Essbase Studio Server runs in the background. The display of server console commands and hints in the server console are blocked; and users are prevented from typing commands in the server console.

The default value is false; however, when Essbase Studio Server is running on UNIX, this property is automatically set to "true" during installation.

Syntax

```
server.runInBackground=true|false
```

Example

```
server.runInBackground=true
```

Note: On UNIX, to run Essbase Studio Server in the foreground, see "Starting Essbase Studio Server in the Foreground on UNIX" on page 54.

server.essbase.blindShare

When set to "true," the `server.essbase.blindShare` property enables duplicate members to be added as shared members.

Also, when set to "true," these special cases for adding duplicate members as shared members are allowed:

- When the duplicate member is not coming from same column as the primary member
- When the duplicate member comes from a non-level zero class (but it is at level zero in the outline)

The default value is false.

To add duplicate members as shared members, add an entry for the `server.essbase.blindShare` property to the `server.properties` file.

Syntax

```
server.essbase.blindShare=true|false
```

Example

```
server.essbase.blindShare=true
```

oracle.jdbc.ReadTimeout

Read timeout while reading from the socket. Timeout is in milliseconds.

If you are using Oracle data sources and are experiencing frequent timeouts, you can increase the value of this option by adding an entry to the `server.properties` file.

The default value is 600000 milliseconds.

Syntax

oracle.jdbc.ReadTimeout=*milliseconds*

Example

```
oracle.jdbc.ReadTimeout=900000
```

data-source-type.cache.size

The internal cache of physical connections in the connection pool. The internal cache consists of physical connections that are always open. Some or all of the connections may be in use concurrently.

The `data-source-type.cache.size` property, together with the `data-source-type.pool.maxsize` property, controls connection pooling for data source connections. Connection pools allow several tasks to query the same data source connection concurrently.

The default size of the cache of connections is 5.

See also "*data-source-type*.pool.maxsize" on page 42.

Syntax

`data-source-type.cache.size=number of physical connections in connection pool`

Valid values for `data-source-type` are:

- Oracle—`oracle`
- IBM DB2—`db2`
- Microsoft SQL Server—`sqlserver`
- MySQL—`mysql`
- Netezza—`netezza`
- Oracle BI EE—`obiee`
- Teradata—`teradata`

Examples by Data Source Type

```
oracle.cache.size=15

db2.cache.size=10

sqlserver.cache.size=15

mysql.cache.size=20

netezza.cache.size=25

obiee.cache.size=5

teradata.cache.size=15
```

data-source-type.pool.maxsize

The maximum number of connections in the connection pool. If the number of connections required exceeds the number of connections specified in `data-source-type.cache.size`, Essbase Studio Server opens temporary connections until the value specified in `data-source-type.pool.maxsize` is reached. As the number of connections required decreases, the temporary connections are destroyed.

The *data-source-type*.pool.maxsize property, together with the *data-source-type*.cache.size property, controls connection pooling for data source connections. Connection pools allow several tasks to query the same data source connection concurrently.

The default for the maximum number of connections in the connection pool is 10.

See also "*data-source-type*.cache.size" on page 42.

Syntax

`data-source-type.pool.maxsize=maximum size of the connection pool`

Valid values for *data-source-type* are:

- Oracle—oracle
- IBM DB2—db2
- Microsoft SQL Server—sqlserver
- MySQL—mysql
- Netezza—netezza
- Oracle BI EE—obiee
- Teradata—teradata

Examples by Data Source Type

```
Oracle.pool.maxsize=20

db2.pool.maxsize=20

sqlserver.pool.maxsize=20

mysql.pool.maxsize=20

netezza.pool.maxsize=20

obiee.pool.maxsize=10

teradata.pool.maxsize=20
```

server.essbase.uniqueMemberFromCaptionBinding

When set to true, while loading members into Essbase, unique member names come from the caption binding expression. The default is false.

If the key and caption bindings are different, and uniqueMemberFromCaptionBinding is set to false, then for drill-through reports to work properly, the **Duplicate member name support** check box must be selected in the **General** tab of the **Essbase Model Properties** dialog box.

Syntax

`server.essbase.uniqueMemberFromCaptionBinding=true|false`

Notes

With `server.essbase.uniqueMemberFromCaptionBinding=false`, when deploying cubes, this is the default behavior for loading members into Essbase:

- For unique member name outlines, member names come from the key binding expression.
- For duplicate member name outlines, member names come from the caption binding expression.
- For unique and duplicate member name outlines:
  - Member name transformation (such as prefix/suffix) works as specified
  - Data load optimization (alias optimizaton) works as specified

When `server.essbase.uniqueMemberFromCaptionBinding=true`, and the caption and key bindings of a dimension element are different, drill-through reports cannot be run from cubes that contain this dimension element.

Example

```
server.essbase.uniqueMemberFromCaptionBinding=true
```

Server Properties File Examples

The following examples are presented:

- "Oracle Example" on page 44
- "IBM DB2 Example" on page 45
- "Microsoft SQL Server Example" on page 45

Oracle Example

When using an Oracle ID (SID), you must append the Oracle SID to the `catalog.url` parameter, using the following syntax:

```
catalog.url=oracle\://host\:port\:OracleSID
```

When using an Oracle Service Name, you must append the Oracle Service Name to the `catalog.url` parameter, using the following syntax:

```
catalog.url=oracle\://host\:port/OracleServiceName
```

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

```
catalog.url=oracle\://sequoia.xyzco.com\:1521\:bpm
server.css.URL=http\://spruce.xyzco.com\:28080/interop/framework/getCSSConfigFile
catalog.db=esbstudio
catalog.username=root
catalog.password=A627FC9A6DEA834C1FA777217871D09E
server.hss.bpmApplication=BPM\:29696
server.datafile.dir=./data/flatfiles
server.essbase.streamingCubeBuilding=true
```

```
server.timeoutPeriod=7200
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
oracle.pool.maxsize=20
oracle.cache.size=15
server.essbase.uniqueMemberFromCaptionBinding=true
```

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:

```
catalog.url=db2\://cypress\:50000;databaseName=BPM_Db
server.css.URL=http\://aspen2.xyzco.com\:28080/interop/framework/getCSSConfigFile
catalog.db=bpm.ROOT
catalog.username=root
catalog.password=A627FC9A6DEA834C1FA777217871D09E
server.hss.bpmApplication=BPM\:29696
server.datafile.dir=$USER_HOME/data/FlatFileDir
server.essbase.streamingCubeBuilding=true
server.timeoutPeriod=7200
server.queueSize=250
server.threadCount=50
server.resourceCount=35
server.sql.fetchSize=500
server.tempDir=$USER_HOME/Temp
server.readLockTimeOut=150
server.writeLockTimeOut=150
db2.pool.maxsize=20
db2.cache.size=15
server.essbase.uniqueMemberFromCaptionBinding=true
```

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:

```
catalog.url=sqlserver\://aspen3.us.oracle.com\:1433
server.css.URL=http\://pine4.us.oracle.com\:28080/interop/framework/getCSSConfigFile
catalog.db=esbstudio.root
catalog.username=root
catalog.password=A627FC9A6DEA834C1FA777217871D09E
server.hss.bpmApplication=BPM\:29696
server.datafile.dir=C:/FlatFileSource
server.essbase.streamingCubeBuilding=true
server.timeoutPeriod=7200
server.queueSize=200
server.threadCount=35
server.resourceCount=25
server.sql.fetchSize=500
server.tempDir=C:/Windows/Temp
server.charset=windows-1251
server.readLockTimeOut=150
server.writeLockTimeOut=150
sqlserver.pool.maxsize=20
sqlserver.cache.size=15
server.essbase.uniqueMemberFromCaptionBinding=true
```

Configuring Logging

Essbase Studio logging uses the Oracle Diagnostic Logging (ODL) framework.

The following Essbase Studio server properties related to logging are deprecated starting with release 11.1.2:

- `logger.file`
- `logger.limit`
- `logger.count`
- `com.hyperion.cp.handlers=com.hyperion.cp.util.LoggerFileHandler`

The property, `com.hyperion.cp.level`, is now set in the `logging.xml` file, which is part of ODL; it is no longer set in the Essbase Studio `server.properties` file.

For information on configuring logging for Essbase Studio and other EPM System products, see the *Oracle Hyperion Enterprise Performance Management System Installation and Configuration Troubleshooting Guide*, available on the Oracle Technology Network.

Working with the Essbase Studio Server Startup File

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

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

The start up file is installed by default in:

```
MIDDLEWARE_HOME/user_projects/epmsystem1BPMS/bpms1/bin
```

The file is named `startBPMS_bpms1_Server.bat` on Windows; `startBPMS_bpms1_Server.sh` on UNIX.

Example

An example of the `startBPMS_bpms1_Server.bat` file on a Windows 2003 operating system is shown below:

```
pushd "C:\Oracle\Middleware\user_projects\epmsystem1\bin"
call setEnv.bat
set JAVA_OPTIONS=-DESSBASE_STUDIO_INSTANCE="C:\Oracle\Middleware\user_projects
   \epmsystem1/BPMS\bpms1" -DsuppressAPSProductInfo=true
set JAVA_HOME=C:\Oracle\Middleware\jdk160_11\jre
call %EPM_ORACLE_HOME%/products/Essbase/EssbaseStudio/Server/startServer.bat
```

Configuring ODBC on UNIX Systems

In Windows environments, you use the ODBC Administrator to configure ODBC connections to data sources. ODBC Administrator is not available on UNIX systems, so you must perform manual configuration. Configuring ODBC on AIX, HP-UX, Solaris, and Linux requires that you complete these procedures:

- "Creating a Symbolic Link to .odbc.ini" on page 48

  For all databases that will be used as data sources in Essbase Studio, you create the symbolic link to `.odbc.ini`.

- "Adding Driver Descriptors to odbcinst.ini" on page 48

  If you are using MySQL Enterprise, Netezza, or Teradata as data sources on UNIX, you must install the ODBC driver and add an entry for the driver to `odbcinst.ini`. For the other supported data sources, the ODBC drivers are installed and configured in `odbcinst.ini`.

- "Adding DSNs to the odbc.ini File" on page 49

  For all databases that will be used as data sources in Essbase Studio, you must configure the ODBC connection in `odbc.ini`.

Note: **Oracle BI EE users:** Instructions for ODBC configuration for UNIX and Linux are included in the *Oracle Hyperion Enterprise Performance Management System Installation and Configuration Guide*.

Note: **MySQL users:** Oracle supports MySQL Enterprise version. Oracle does not support MySQL Community version.

Note: Essbase Studio does not support data source table names and column names that contain spaces or special characters, such as a period (.). See Appendix C, "Naming Restrictions for Essbase Studio" for a complete listing of unsupported characters.

Flat File Data Sources: In Essbase Studio, ODBC configuration is not required for flat file data sources.

Creating a Symbolic Link to .odbc.ini

➤ To create a symbolic link for `.odbc.ini`:

1. **Create a symbolic link to the `$ESSBASEPATH/bin/.odbc.ini` from the UNIX user home directory; for example:**

   ```
   ln -s $EPM_ORACLE_HOME/common/ODBC/Merant/x.x/odbc.ini /home/myfolder/.odbc.ini
   ```

2. **For MySQL, Netezza, and Teradata:** Continue with "Adding Driver Descriptors to odbcinst.ini" on page 48.

Adding Driver Descriptors to odbcinst.ini

For ODBC drivers that are installed with the EPM System, driver descriptors are automatically added to `odbcinst.ini`. For drivers that are not installed by the EPM System, you must manually add the driver descriptor to `odbcinst`.

➤ **For MySQL, Netezza, and Teradata:** To add the driver descriptor to `odbcinst.ini`:

1. **Locate the appropriate `odbcinst.ini`.**

   32-bit:

   $EPM_ORACLE_HOME`/common/ODBC/Merant/6.1/odbcinst.ini`

   64-bit:

   $EPM_ORACLE_HOME`/common/ODBC-64/Merant/6.1/odbcinst.ini`

2. **Add the appropriate driver descriptor entry.**

   For example:

   MySQL

   ```
   [ODBC Drivers]
   MySQL ODBC 3.51 Driver =Installed

   ...

   [MySQL ODBC 3.51 Driver]
   Description = ODBC 3.51 for MySQL
   DRIVER = /usr/lib/libmyodbc3.so
   SETUP = /usr/lib/libmyodbc3S.so
   UsageCount = 2
   ```

   Netezza

   ```
   [ODBC Drivers]
   Netezza=Installed

   [Netezza]
   ```

```
Driver=/usr/local/nz/lib/libnzodbc.so
Setup=/usr/local/nz/lib/libnzodbc.so
```

Teradata

```
[ODBC Drivers]
Teradata=Installed

[Teradata]
Driver=/usr/odbc/drivers/tdata.so
Setup=/usr/odbc/drivers/tdata.so
```

3 **Save the file and continue with "Adding DSNs to the odbc.ini File" on page 49.**

Adding DSNs to the odbc.ini File

You configure data source names (DSNs) for data source databases in an `odbc.ini` file. In one section of the file, add a name and description for the ODBC data source. In a separate, newly created section of the file, provide the ODBC driver path, file name, and all other required driver settings.

The EPM System installation program installs a sample `odbc.ini` file in the `products/common/ODBC/Merant/6.1` and `products/common/ODBC-64/Merant/6.1` directory. The file contains generic ODBC connection and configuration information for supported ODBC drivers. Use the file as a starting point to map the ODBC drivers that you use to the data source databases.

Tip: The procedure in this section shows you how to configure a DSN by manually editing the `odbc.ini` file.

➤ To add an ODBC data source to an `odbc.ini` file:

1 **On the computer where the EPM System common components are installed, locate and open the appropriate `odbc.ini` file.**

Use the vi $ODBCINI command to edit the `odbc.ini` file and statements:

32-bit:

$EPM_ORACLE_HOME`/common/ODBC/Merant/6.1/odbc.ini`

64-bit:

$EPM_ORACLE_HOME`/common/ODBC-64/Merant/6.1/odbc.ini`

2 **Locate the section starting with `[ODBC Data Sources]` and add a new line with the data source name and description; for example:**

```
mydata=DataDirect 6.1 Oracle Wire Protocol
```

3 **Add a new section to the file by creating a new line with the new DSN enclosed in brackets; for example:**

```
[mydata]
```

4 **On the lines following the data source name, add the full path and file name for the ODBC driver required for this data source and any other required database information.**

Use the examples shown in the following sections as guidelines for specific RDBMSs.

Note: Ensure that the ODBC driver file actually exists in the location that you specify for the "Driver=" setting.

5 **When you finish editing `odbc.ini`, save the file and exit the text editor.**

Configuration Verification Tips

- You can run `ivtestlib` (32-bit) or `ddtestlib` (64-bit) to verify that the environment is set to run the correct ODBC driver file.

  For example, run `ivtestlib` and paste the path and file name that follow `Driver=` in the `odbc.ini` file that you edited.

- In the `odbc.ini` file, under the `[ODBC]` heading, verify that these parameters setting are specified as indicated:
  - `InstallDir`—The full path to the driver installation directory. This setting should not contain a variable descriptor, such as `<installDir>`.
  - `TraceDll`—The full path to the `/lib` directory. This setting should not contain a variable descriptor, such as `<traceDll>`.

  Use the examples shown in the following sections as guidelines for specific RDBMSs.

For information about the `odbc.ini` file and the ODBC driver settings for each RDBMS or flat file data source, see the *DataDirect Connect ODBC Reference* in the `products/common/ODBC/Merant/6.0/books` or `products/common/ODBC-64/Merant/6.0` directory. For information about vendor-supplied ODBC driver settings, refer to the installation documentation for the vendor-supplied ODBC drivers.

Flat File Data Sources: In Essbase Studio, ODBC configuration is not required for flat file data sources.

Example of ODBC Settings for Oracle

The following example illustrates how you might edit `odbc.ini` to connect to a data source database named "oradata" on Oracle (on Solaris), using a DataDirect Wire Protocol driver.

```
[ODBC Data Sources]
oradata=DataDirect 6.1 Oracle Wire Protocol

...

[oradata]
Driver=<$EPM_ORACLE_HOME>/common/ODBC/Merant/6.1/lib/ARora25.so
HostName=oraclehost
SID=ORADB
PortNumber=1521
```

Examples of ODBC Settings for IBM DB2 UDB

The following example illustrates how you might edit `odbc.ini` to connect to a data source database named "db2data" on IBM DB2 (on AIX), using a DataDirect Wire Protocol driver.

```
[ODBC Data Sources]
db2data=DataDirect 6.1 DB2 Wire Protocol

...

[db2data]
Driver=<$EPM_ORACLE_HOME>/common/ODBC/Merant/6.1/lib/ARdb225.so
Database=DB2DB
IpAddress=db2host
TcpPort=50000
```

Example of ODBC Settings for MySQL

The following example illustrates how you might edit `odbc.ini` to connect to a data source database named "ODBC_MySQL" on MySQL using a vendor-supplied ODBC driver.

```
[ODBC Data Sources]
ODBC_MySQL=MyODBC 3.51 Driver DSN

...

[ODBC_MySQL]
Driver = /usr/local/lib/libmyodbc3.so
Description = Connector/ODBC 3.51 Driver DSN
SERVER = localhost
PORT = 3306
USER = root
Password =
Database = test
OPTION =
SOCKET =
```

Example of ODBC Settings for Microsoft SQL Server

The following example illustrates how you might edit `odbc.ini` to connect to a data source database named "sqldata" on Microsoft SQL Server 2005 (on HP-UX), using a DataDirect Wire Protocol driver.

```
[ODBC Data Sources]
sqldata=DataDirect 6.1 SQL Server Wire Protocol

...

[sqldata]
Driver=<$EPM_ORACLE_HOME>/common/ODBC/Merant/6.1/lib/ARsqls25.so
Database=SQLDB
Address=mssqlhost,1433
EnableQuotedIdentifiers=1
```

Examples of ODBC Settings for Netezza

The following example illustrates how you might edit `odbc.ini` to connect to a data source database named "NZSQL" on Netezza using a vendor-supplied ODBC driver.

```
[ODBC Data Sources]
NZSQL = NetezzaSQL

...

[NZSQL]
Driver = /usr/local/nz/lib/libnzodbc.so
Description = NetezzaSQL ODBC
Servername = <123.4.5.6>
Port = 5480
Database = system
Username = admin
Password = password
ReadOnly = false
ShowSystemTables = false
LegacySQLTables = false
LoginTimeout = 0
QueryTimeout = 0
DateFormat = 1
NumericAsChar = false
SQLBitOneZero = false
StripCRLF = false
```

Example of ODBC Settings for Teradata

The following example illustrates how you might edit `odbc.ini` to connect to a data source database named "terasource" on Teradata using a vendor-supplied ODBC driver.

Solaris and AIX

```
[ODBC Data Sources]
terasource=Teradata data source

...

[terasource]
Driver=/usr/odbc/drivers/tdata.so #teradata installation path of .so file
DBCName=abcd0072.us.xyzco.com
PortNumber=1025
```

HP-UX

```
[ODBC Data Sources]
terasource=Teradata data source

...

[terasource]
Driver=/usr/odbc/drivers/tdata.sl #teradata installation path of .sl file
DBCName=abcd0072.us.xyzco.com
PortNumber=1025
```

Configuring JDBC Drivers

During cube deployment, when Essbase Studio is run in streaming mode, Essbase Studio Server uses JDBC drivers to query the external data source directly.

The JDBC drivers for Oracle, Oracle BI EE, IBM DB2, and Microsoft SQL Server are installed when you install Essbase Studio.

For MySQL, Netezza, and Teradata, you must obtain the JDBC driver from the manufacturer and install it, as described in the following sections.

Note: Perform the tasks in this section for the applicable JDBC driver after you have installed and configured Essbase Studio.

MySQL

➤ To set up the MySQL JDBC driver:

1. **From the MySQL Web site, download the appropriate version of the file to this directory:**

   `EPM_ORACLE_HOME/products/Essbase/EssbaseStudio/Server`

   The name of the driver file you download will vary depending on the version downloaded.

2. **Rename the file** `mysql-connector-java.jar`**.**

   Essbase Studio will not recognize the driver file until it is renamed.

Netezza

➤ To set up the Netezza JDBC driver:

1. **From the Netezza Web site, download the appropriate version of the file to this directory:**

   `EPM_ORACLE_HOME/products/Essbase/EssbaseStudio/Server`

   The name of the driver file you download will vary depending on the version downloaded.

2. **Rename the file** `nzjdbc3.jar`**.**

   Essbase Studio will not recognize the driver file until it is renamed.

Teradata

➤ To set up the Teradata drivers:

1. **From the Teradata Web site, download the Teradata JDBC driver library files** `tdgssconfig.jar` **and** `terajdbc4.jar`**.**

2. **Extract the files to the this directory:**

   `EPM_ORACLE_HOME/products/Essbase/EssbaseStudio/Server`

   Note: Consult Teradata directly for any compatibility issues or questions.

Starting and Stopping Essbase Studio Server and Console

After you have completed configuration using the Oracle's Hyperion Enterprise Performance Management System Configurator, and performed any configuration in the `server.properties` and `EASLaunch.properties` files, you are ready to start the Essbase Studio Server and Console.

Information on starting and stopping all EPM System products, including Essbase Studio Server and Console, is located in the *Oracle Hyperion Enterprise Performance Management System Installation and Configuration Guide*, and can be found in the EPM System Documentation Library on Oracle Technology Network (OTN):

http://www.oracle.com/technology/documentation/epm.html

If you plan to run Essbase Studio Server in the foreground on UNIX, see "Starting Essbase Studio Server in the Foreground on UNIX" on page 54.

Starting Essbase Studio Server in the Foreground on UNIX

By default, Essbase Studio Server runs in the background on UNIX. To run Essbase Studio Server in the foreground, edit the following:

- The server property, `server.runInBackground`.
- These environment variables:
  - EPM_ORACLE_HOME=<value>
  - EPM_ORACLE_INSTANCE=<value>
  - JAVA_HOME=$EPM_ORACLE_HOME/../jdk160_11/jre
  - JAVA_OPTIONS=-DESSBASE_STUDIO_INSTANCE\= $EPM_ORACLE_INSTANCE/BPMS/bpms1 -DsuppressAPSProductInfo=<T|F>
- The Essbase Studio startup shell script, `startServer.sh`.

➤ To start Essbase Studio Server in the foreground on UNIX:

1 **In the Essbase Studio `server.properties` file, set the `server.runInBackground` property to "false" or comment it out.**

   This property is set to "true" by default.

   The `server.properties` file is located in `Oracle/Middleware/user_projects/epmsystem1/BPMS/bpms1/bin/server.properties`.

   Note: See "server.runInBackground" on page 40 for information on this property.

2 **Set these variables as shown in the environment where you will be running `startServer.sh`:**

   ```
   EPM_ORACLE_INSTANCE=/<install path>/Oracle/Middleware/user_projects/epmsystem1
   ```

```
EPM_ORACLE_HOME=/<install path>/Oracle/Middleware/EPMSystem11R1

JAVA_HOME="${EPM_ORACLE_HOME}/../jdk160_11/jre"

JAVA_OPTIONS="-DESSBASE_STUDIO_INSTANCE=${EPM_ORACLE_INSTANCE}/BPMS/bpms1
-DsuppressAPSProductInfo=true"
```

3 **Edit the Essbase Studio `startServer.sh` shell as follows.**

The `startServer.sh` file is located in `$EPM_ORACLE_HOME/products/Essbase/EssbaseStudio/Server/startServer.sh`.

a. Locate the last line of the file:

```
nohup "${JAVA_HOME}/bin/java" -Xms128m -Xmx768m $JAVA_OPTIONS
  -jar "${EPM_ORACLE_HOME}/products/Essbase/EssbaseStudio/
  Server/server.jar" >/dev/null &
```

b. Remove “nohup” from the beginning of the line, the STDOUT to null direction (>/dev/null), and the background processing command (&) from the line; for example:

```
"${JAVA_HOME}/bin/java" -Xms128m -Xmx768m $JAVA_OPTIONS -jar "${EPM_ORACLE_HOME}
  /products/Essbase/EssbaseStudio/Server/server.jar"
```

4 **Start Essbase Studio Server by running the following statement:**

```
./startServer.sh
```

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 11.1.2.0.00 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 35 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

  When you run this command, you are prompted to enter the new queue size.

  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 28.

- **`sthd`**—Sets the number of worker threads allocated to Essbase Studio Server.

  See "server.threadCount" on page 35 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 28.

- **`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 28.

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

  Command

  exit

  Example

  `exit`

Exporting and Importing the Essbase Studio Catalog Database

Essbase Studio provides a mechanism for exporting the catalog database, or selected catalog database artifacts, to an XML file. The XML file can then be imported to another computer to recreate the catalog. For more information, see:

- "About Catalog Export and Import" on page 58
- "Exporting the Catalog Database to an XML File" on page 59
- "Importing the Catalog XML File" on page 61

About Catalog Export and Import

You may need to move the Essbase Studio catalog from one computer to another. Essbase Studio enable you to export the catalog database into a file, which can then be imported to a different machine in order to recreate the catalog.

The process begins with the export operation. Using the Export option on the File menu, you may perform an action:

- Export the entire catalog
- Export individual catalog artifacts, such as data source connections, dimension elements, hierarchies, alias sets, cube schemas, Essbase models, and drill-through reports.

You may also right-click individual or multiple catalog artifacts in the **Data Source Navigator** or the **Metadata Navigator** and then select the Export option from the shortcut menu.

No matter which export method you choose, the catalog database or selected artifacts are exported into an XML file.

You then import the XML file at the new location or machine. You must import the entire contents of the file; you cannot selectively import artifacts from the file.

Notes

- When specific catalog artifacts are exported, the XML file that is created contains the descendants of the selected catalog artifact. For example, if a hierarchy is selected, the dimension elements and data source connection in the lineage of that hierarchy are also exported.

- When exporting data source connections, the password to the source database is not included in the resulting XML file. After importing the XML file, add the password back by editing the connection, as described in "Editing Data Source Connection Properties" on page 99.

Exporting the Catalog Database to an XML File

You can export the entire catalog database or only selected catalog elements. See the following topics for instructions:

- "Exporting the Entire Catalog Database" on page 59
- "Exporting Selected Catalog Elements" on page 60

Exporting the Entire Catalog Database

➤ To export the Essbase Studio catalog database to an XML file:

1. **In Essbase Studio Console, select File, and then Export.**
2. **In the Export dialog box, click Browse and navigate to the location where you will place the exported catalog XML file.**
3. **In File Location, enter a name for the XML file.**
4. **Optional: In File Description, enter descriptive text.**
5. **Under File Contents, choose an option:**
   - **Export entire catalog**

     To export the entire catalog, continue this procedure with step 6.
   - **Export partial catalog**

     To export selected elements of the catalog, see "Exporting Selected Catalog Elements" on page 60.
6. **Click OK to complete the export.**

   The entire catalog database is exported to an XML file with the name that you specified in step 3.

   Note: If errors occurred during export, the error details are displayed in a message box.
7. **Verify that the catalog database XML file exists in the location that you specified step 2.**

You can now import the XML catalog database to the new location. See "Importing the Catalog XML File" on page 61.

Exporting Selected Catalog Elements

➤ To export selected catalog elements (partial catalog) to an XML file:

1 **Perform an action:**

- Complete step 1 through step 5 in "Exporting the Catalog Database to an XML File" on page 59, selecting **Export partial catalog** in step 5.
- Right-click on elements in the **Metadata Navigator** or data sources in the **Data Source Navigator** and select **Export** from the shortcut menu.

The **Export** dialog box is displayed with the **Export partial catalog** option selected and the **Add** button enabled.

2 **Click Add and select an option:**

- **Add Metadata**

  The **Select Element to Export** dialog box displays the contents of the **Metadata Navigator**. Perform these steps:

  a. Navigate to the items that you want to export.

  b. When you have finished making selections, click **OK**.

- **Add Data Source**

  The **Select Data Source to Export** dialog box displays the contents of the **Data Source Navigator**. Perform these steps:

  a. Select the data sources that you want to export.

  b. When you have finished making selections, click **OK**.

The metadata elements and data sources that you have selected for export are listed in the table in the **File Contents** portion of the **Export** dialog box.

The full path of each element is shown under "Names." The type of element—such as connection or hierarchy—is shown under "Type."

3 **Optional: To delete any items listed before performing the export, select the item in the File Contents table and click Delete,** .

4 **Click OK.**

The selected items from the catalog are exported to the XML file that you designated in step 2 and step 3 of "Exporting the Catalog Database to an XML File" on page 59.

Note: If you encounter an out-of-memory error during export, increase the virtual memory setting for Essbase Studio Console, as described in "Configuring Virtual Memory" on page 62.

You can now import the XML catalog database artifacts to the new location. See "Importing the Catalog XML File" on page 61.

Importing the Catalog XML File

Preparing to Import

➤ Before you begin the catalog XML file import procedure, complete the following tasks:

1. **Using the method of your choice, move the catalog XML file that you created in "Exporting the Catalog Database to an XML File" on page 59 or "Exporting Selected Catalog Elements" on page 60 to the location or computer where you will be importing it.**

   For example, you may use an FTP program to move the file to the target location on another computer.

2. **Ensure that the `catalog.db` property in `server.properties` specifies the location of the target catalog database.**

   See "catalog.db" on page 32.

   For a new catalog, create an empty database to serve as the target catalog database using the `catalog_schema.sql` scripts included in the Essbase Studio installation. The scripts are located in:

   EPM_ORACLE_HOME`/products/Essbase/EssbaseStudio/server/database/common/`*your_RDBMS*

   Note: This task assumes that the catalog RDBMS program is installed on the target machine.

Importing

Be sure to complete the tasks in Preparing to Import before beginning this procedure.

➤ To import the Essbase Studio catalog database XML file:

1. **In Essbase Studio Console, select File, and then Import.**
2. **In the Import dialog box, click Browse, and in Import File, navigate to the location of the exported catalog XML file.**
3. **Select the XML file to import, and click Open to return to the Import dialog box.**

   Note that the **File Name**, **File Description**, and **File Content Type** fields are populated with the information that you entered during the export process.

4. **In the Options for importing into catalog group, select an option:**
   - **Check for elements and overwrite them**—If the element already exists in the catalog database, it is overwritten with the new element from the XML file.
   - **Check for elements but do not overwrite them**—If the element already exists in the catalog database, it is retained; the duplicate element in the XML file is not used.
   - **Do not check for elements; all elements are new**—A catalog database is created using the XML file.
5. **Click OK to import the elements in the XML file into the catalog database.**

The XML file populates the catalog database that is specified by `catalog.db` in the `server.properties` file (see "catalog.db" on page 32).

If errors occur during import, an error message box containing the details is displayed.

Note: If you encounter an out-of-memory error during import, increase the virtual memory setting for Essbase Studio Console, as described in "Configuring Virtual Memory" on page 62.

Configuring Virtual Memory

Some operations, such as the export of large catalog databases or a very complex cube deployment, may push the virtual memory limits of the Essbase Studio Console. Use the following procedure to modify the virtual memory setting.

➤ To configure the virtual memory of Essbase Studio Console:

1 **Open `EssbaseStudio.ini`, in `EPM_ORACLE_HOME/products/Essbase/EssbaseStudio/Console`.**

By default, the `EssbaseStudio.ini` file contains these two lines:

```
-vmargs
-Xmx1024m
```

The second line contains the virtual memory setting. By default, virtual memory is set for 1024 MB.

2 **Increase memory by increasing the virtual memory setting; for example:**

```
-Xmx1536m
```

The example above specifies 1536 MB of virtual memory.

3 **Save the modified `EssbaseStudio.ini` file.**

4 **Restart the Essbase Studio Console.**

Updating References to Rehosted EPM System Products

Whenever you move EPM System products to a new host machine (for example, after upgrade) you must update references for some products to reflect the new host name and port number.

Note: Always perform the rehosting steps for Essbase before performing rehosting steps for any other EPM System product, including Essbase Studio.

- For rehosted Essbase Server instances or clusters, use the procedure in "Updating References to a Rehosted Essbase Server" on page 63.

- For rehosted Essbase Studio Server instances, use the procedure in "Updating Cube Linkage" on page 276.
- For rehosted Performance Management Architect data sources, use the procedure in "Editing Data Source Connection Properties" on page 99

See the *Oracle Hyperion Enterprise Performance Management System Installation and Configuration Guide* for complete information on EPM System product rehosting.

About Updating References to a Rehosted Essbase Server

Your Essbase Server connections and deployed cubes are affected by changes in host, port, or data encryption information for the underlying Essbase Server instances or clusters.

You update the references to the "rehosted" Essbase Server instances and clusters in the **Rehost Essbase Connections** dialog box. Provide the new host or cluster, port, and, if required, data encryption method, as described in "Updating References to a Rehosted Essbase Server" on page 63.

Note: Always perform the rehosting steps for Essbase before performing rehosting steps for any other EPM System product, including Essbase Studio.

Other Rehosting Topics

- For rehosted Essbase Studio Server instances, use the procedure in "Updating Cube Linkage" on page 276.
- For rehosted Performance Management Architect data sources, use the procedure in "Editing Data Source Connection Properties" on page 99

See the *Oracle Hyperion Enterprise Performance Management System Installation and Configuration Guide* for complete information on EPM System product rehosting.

Updating References to a Rehosted Essbase Server

Note: Always perform the rehosting steps for Essbase before performing rehosting steps for any other EPM System product, including Essbase Studio.

➤ To updated references to a rehosted Essbase Server instance or cluster:

1. In Essbase Studio Console, select **Tools**, and then **Rehost Essbase Connections.**
2. In **Rehost Essbase Connections**, select the connection to rehost under **Essbase connections.**

   The old cluster name or Essbase host and port number, and, if applicable, data encryption method, are displayed under **Host/Port/Encryption in Catalog**.

3 **Under New Host/Port/Encryption, enter the new cluster name or new host name, port number, and, if applicable, data encryption method.**

Note: For Essbase Server clusters, only the cluster name is required. The port number is not required. By default, the new cluster name is displayed in the **New Host/Port/Encryption** column.

Use the following syntax:

- For an Essbase Server instance, no data encryption, enter:

  `host:port`

- For an Essbase Server instance, with data encryption, enter:

  `host:port:ssl`

- For an Essbase Server cluster, no data encryption, enter:

  `cluster_name`

4 **Optional: Click Test Connection to validate the entry that you made under New Host/Port/Encryption.**

Note: Essbase Studio does not validate entries as you enter them. You must click **Test Connection** to validate the information that you entered in step 3.

5 **Select an Update Deployment History option for the currently selected Essbase Server connection:**

- **Update the host name and port number for all deployment history**—Replaces all old Essbase Server host and port references to the new host and port that you specified in step 3.

  This is the default option.

- **Replicate the last successful deployment history and update the copy only**—Makes a copy of the last successful deployment history listing, and then updates the copy with the new host and port information, and the date and time of the rehosting.

  Tip: When selecting the "Replicate" option, after the update is complete, each rehosted Essbase model displays twice in the **Metadata Navigator**: once with the old host:port information and once with the new host:port information.

Note: Deployment history is updated only for successfully rehosted Essbase Server connections.

6 **Click Update.**

7 **For the rehosting status of each Essbase Server connection that is rehosted, check the Update Status column in the Rehost Essbase Connections dialog box.**

If the rehost is successful, the **Host/Port/Encryption in Catalog** column is updated with the new cluster or host and port information and, if applicable, encryption information. If the rehost fails, an error message is displayed.

Note: If other Essbase Server connections have the same old host:port setting, rehosting one instance rehosts all other instances.

3 User Management and Security

In This Chapter

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. 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 the 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 roles 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 Roles* appendix in the *Oracle Hyperion Enterprise Performance Management System User and Role Security 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 *Launching Shared Services Console* topic in the *Oracle Hyperion Enterprise Performance Management System User and Role Security 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. Users also require an Essbase Server role such as Create/Delete Application or Administrator.

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, see the *Provisioning Users and Groups* topic and the *Provisioning Essbase* chapter in the *Oracle Hyperion Enterprise Performance Management System User and Role Security Guide*.

4 Data Source Connections

In This Chapter

Overview

In Essbase Studio, you connect to various types of data sources using the **Connection Wizard**. Data sources can be relational databases, Oracle BI EE sources, Performance Management Architect applications, 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 72.

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 72.
- Optionally, select which source tables or files to include in the connection.

  See "Selecting Tables to Include in the Connection" on page 73.
- Select whether to create a minischema for this data source.

  See "Selecting a Minischema Option" on page 73.

  If you choose to create a minischema, you will also populate it.

  See "Populating a Minischema" on page 73.
- Create metadata elements from the data source elements.

  See "Creating Metadata Elements" on page 74.

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: After defining the data source parameters in the first page of the **Connection Wizard**, completing the remaining wizard pages is optional.

Defining Connection Parameters

You define connection parameters on the first page of the **Connection Wizard**. The parameters that 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, MySQL, Netezza, Oracle BI EE (Presentation Layer or Business Model option), Oracle Real Application Clusters (Oracle RAC), and Teradata. See "Defining Connection Parameters for Relational Sources" on page 75.

- Essbase Server instances—See "Creating Connections to Essbase" on page 83.
- Dimension Server sources—Used for Performance Management Architect sources; see "Creating Connections to Performance Management Architect Data Sources" on page 84.
- Flat or text file data sources—See "Defining Connection Parameters for Text File Sources" on page 87.

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, MySQL, Netezza, Oracle BI EE, Oracle RAC, and Teradata. See "Defining Connection Parameters for Relational Sources" on page 75.

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 BI EE, 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, MySQL, Netezza, Oracle BI EE, Oracle RAC, and Teradata. See "Selecting a Minischema Option for Relational Sources" on page 79.
- Essbase Server sources—See "Creating Connections to Essbase" on page 83.
- Dimension Server sources—Includes Performance Management Architect sources; see "Creating Connections to Performance Management Architect Data Sources" on page 84.
- Flat or text file data sources—See "Selecting a Minischema Option for Text File Sources" on page 90.

Populating a Minischema

When you are creating a minischema during the data source connection process, you 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.

- Relational data sources—Includes relational sources such as Oracle, IBM DB2, Microsoft SQL Server, MySQL, Netezza, Oracle BI EE, Oracle RAC, and Teradata. See "Populating a Minischema for Relational Sources" on page 80.

- Flat or text file data sources—See "Populating a Minischema for Text File Sources" on page 91.

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—Includes relational sources such as Oracle, IBM DB2, Microsoft SQL Server, MySQL, Netezza, Oracle BI EE (Presentation Layer only), Oracle RAC, and Teradata. See "Creating Metadata Elements for Relational Sources" on page 81.

  For Oracle BI EE (Business Model only), see "Creating Oracle BI EE Dimensions" on page 81.
- Dimension Server sources—Includes Performance Management Architect sources. See "Creating Metadata Elements for Performance Management Architect Sources" on page 86.
- Flat or text file data sources—See "Creating Metadata Elements from Text File Sources" on page 92.

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
- Oracle RAC
- IBM DB2
- Microsoft SQL Server
- MySQL
- Netezza
- Teradata

You may also connect to Oracle BI EE sources as you would relational sources, as either the Presentation Layer or Business Model.

- For Presentation Layer connections, you may not see all the corresponding physical tables in the source. You may see views instead.
- For Business Model connections, you see the OBI dimensions.

Note: MySQL, Netezza, and Teradata can be used as a data source if you have installed the appropriate JDBC drivers. See "Configuring JDBC Drivers" on page 53.

These topics cover the workflow for creating a connection to a relational data source:

- "Defining Connection Parameters for Relational Sources" on page 75.
- "Selecting Tables for Relational Sources" on page 77.
- "Selecting a Minischema Option for Relational Sources" on page 79.
- "Populating a Minischema for Relational Sources" on page 80.
- "Creating Metadata Elements for Relational Sources" on page 81.

  If you are connecting to an Oracle BI EE source, see "Creating Oracle BI EE Dimensions" on page 81.

Defining Connection Parameters for Relational Sources

➤ To define the parameters of a data source:

1 In the **Data Source Navigator**, right-click **Data Sources** in the physical tree, select **New**, and then **Data Source.**

  The **Define Parameters** page of the **Connection Wizard** is displayed.

2 Enter a **Connection Name.**

3 Enter an optional **Description.**

4 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.

5 **Oracle BI EE users:** Choose an Oracle BI EE layer for this data source connection:

  - **Presentation Layer**—A customized view of metadata in the business model.
  - **Business Model**—The mapping of objects in the physical data source to metadata.

6 **Oracle RAC users:** Click the **Add RAC server node** button and provide the following information for at least one node in your Oracle RAC connection.

  - **Service Name**—Enter a new Oracle service name; or select a previously-entered service name from the drop-down list.

- **Server name**—Enter a new Oracle RAC server name; or select a previously-entered server name from the drop-down list.
- **Port**—Enter a new Oracle RAC port number; or select a previously-entered port number from the drop-down list.

Repeat this step for all Oracle RAC server nodes that you want to add to the data source connection.

7 In **Server Name**, enter the name of the server where the database resides, or select a previously-entered server name from the drop-down list.

Note: This step does not apply to Oracle RAC users.

8 **Oracle users:** 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.

9 **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.

10 Enter the **User Name** and **Password** for the selected data source.

11 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 7.

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

12 **IBM DB2 users only:** In **Authentication Method**, select an option:

- **No Encryption**
- **Encrypt Password**
- **Encrypt UserID and Password**
- **Client**

Note: When you specify an authentication method other than "No Encryption" for a data source, then you must perform cube deployments in "streaming" mode. Set the `server.essbase.streamingCubeBuilding` property to "true" to enable deployments in streaming mode. See "server.essbase.streamingCubeBuilding" on page 34 for more information on this property.

13 **Optional:** Modify the **Connection Pool** settings.

Pool Max Size, together with **Cache Size**, controls connection pooling for data source connections. Connection pools allow several tasks to query the same data source connection concurrently.

a. In **Pool Max Size**, enter a number to change the maximum number of connections in the connection pool.

Pool Max Size specifies the maximum number of connections in the connection pool. If the number of connections required exceeds the number of connections specified in **Cache Size**, Essbase Studio Server opens temporary connections until the value specified in **Pool Max Size** is reached. As the number of connections required decreases, the temporary connections are destroyed.

To make changes in the default value shown in **Pool Max Size**, modify the server property, `pool.maxsize`, as described in "*data-source-type*.pool.maxsize" on page 42.

b. In **Cache Size**, enter a number to change the internal cache of physical connections in the connection pool.

The internal cache consists of physical connections that are always open. Some or all of the connections may be in use concurrently.

To make changes in the default value shown in **Cache Size**, modify the server property, `cache.size`, as described in "*data-source-type*.cache.size" on page 42.

14 **Optional:** 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.

15 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 77.

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.

Note: For Oracle BI EE data sources, this procedure is used with Presentation Layer sources only (not with Business Model sources).

➤ 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 **Optional:** Select **Lock catalog during exploration** to prevent other users from writing to the catalog database during the data source creation process.

Note: Because data source creation involves database scraping, you may not want other users adding, modifying, or deleting source or metadata elements until the process completes.

6 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 79.

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 Data Source Connections" on page 93.

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 hierarchy information. If you are accessing a flat or text file data source, the "Use Introspection to Detect Hierarchies" option is not available.

- Skip creation of a minischema diagram.

  See "Skipping Minischema Creation for Relational Sources" on page 79.

- Complete the data source connection process using introspection.

Note: If you are accessing an Oracle BI EE data source, you may create a minischema, but it can contain only self joins (joins between columns in a single file).

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 80.

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 81.

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 77 are included by default in the **Tables in schema** list in the Populate Minischema page of the Connection Wizard.

➤ To populate the minischema:

1 **In Populate Minischema, to remove tables from the 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.**

The tables are moved to the **Available Tables** list

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

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

The tables are moved to the **Tables in Schema** 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 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 81.

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.

Note: This procedure applies to Oracle BI EE data sources that are created with the Presentation Layer option selected. If you have created a data source for Oracle BI EE using the Business Model option, see "Creating Oracle BI EE Dimensions" on page 81.

➤ 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.**

   View the new metadata elements listed in the **Metadata Navigator**.

   If you created a minischema, view the new minischema in the minischema work area of the Essbase Studio Console.

Creating Oracle BI EE Dimensions

During Oracle BI EE connection creation, if you selected the Business Model option, Essbase Studio explores the source database for dimensions. From the dimensions that you select, Essbase Studio derives hierarchies, and the dimension elements that are included in the hierarchies, and adds them to the **Metadata Navigator**.

Note: For information on how Essbase Studio generates key and caption binding expressions for the dimension elements that are created, see "Rules for Generating Key and Caption Bindings for Oracle BI EE Business Model Dimension Elements" on page 357.

➤ To create metadata elements that include dimension elements and hierarchies:

1 **In the Create OBIEE dimensions page of the wizard, in the Available OBI dimensions list, select dimensions to create metadata elements, including hierarchies and their related dimension elements.**

- Select the check box next to the data source name at the top of the **Available OBI dimensions** list to create metadata elements from all available OBI dimensions in the data source.
- Select the check box next to only those OBI dimensions for which you want to create metadata elements.

Dimensions you select are added to the folder for this data source in the **Metadata Navigator**.

2 **In Select OBI fact tables, select the tables to designate as fact tables.**

- Select the check box next to the data source name at the top of the **Select OBI fact tables** list to specify all fact tables candidates.
- Select the check box next to only the fact table candidates that you want to specify as fact tables.

Note: When scraping Oracle BI EE version 11.1.1.6 and later sources, the **Select OBI fact tables** panel will list only fact tables. Versions of Oracle BI EE older than 11.1.1.6 will list all tables (regular tables and fact tables).

3 **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**.

4 **Optional: Select the Create cube schema and Essbase model check box.**

Using the Oracle BI EE fact tables and dimensions, you can specify measures and hierarchies to create a cube schema and an Essbase model, as described in "Creating an Oracle BI EE Cube Schema and Essbase Model" on page 83.

5 **Click Next to create a cube schema from the Oracle BI EE elements, or click Finish.**

View the new metadata elements—dimension elements and hierarchies—listed in the **Metadata Navigator**.

If you created a minischema, view the new minischema in the minischema work area of the Essbase Studio Console.

Creating an Oracle BI EE Cube Schema and Essbase Model

If you chose to create a cube schema and Essbase model in step 4 of "Creating Oracle BI EE Dimensions" on page 81, then, in this page of the wizard, you use the Oracle BI EE dimension elements and hierarchies you just created to set up a cube schema and automatically create an Essbase model.

➤ To set up a cube schema and Essbase model using Oracle BI EE fact tables and dimensions:

1 In the **Create OBIEE cube schema** page of the wizard, from the **Available Fact Tables** list, expand the Oracle BI EE fact tables, select the columns to move to the **Measures** list, and click the right arrow.

   You cannot make selections at the fact table level. However, you may select one or more columns from the fact tables listed.

   Note: You can also use drag-and-drop to move dimensions between the **Available Fact Tables** list and the **Measures** list.

2 In the **Available Dimension Elements** list, select the Oracle BI EE dimensions to move to the **Hierarchies** list, and click the right arrow.

   Note: You can also use drag-and-drop to move dimensions between the **Available Dimension Elements** list and the **Hierarchies** list.

3 Click the **Finish** button.

   View the new metadata elements—dimension elements, hierarchies, cube schema, and Essbase model—listed in the **Metadata Navigator**, either in the default folder for the metadata elements or in the folder you specified in step 3 of "Creating Oracle BI EE Dimensions" on page 81.

   You may work with these elements as you would elements created from any other relational source.

Creating Connections to Essbase

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

You may choose the SSL data encryption option, which allows secure communication between Essbase Studio Server and Essbase.

➤ To define a connection to an Essbase instance or cluster:

1 In the **Data Source Navigator**, right-click **Data Sources** in the physical tree, select **New**, and then **Data Source**.

2 Enter a **Connection Name**.

3 **Optional:** Enter a **Description**.

4 In **Data Source Type**, select **Essbase Server**.

5 **Under Parameters, in Essbase Server, enter the name of computer where this Essbase Server instance resides or enter an Essbase Server cluster name.**

Note: Cluster names are case sensitive. The default cluster name is `EssbaseCluster-1`.

6 **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.

7 **Optional: If you are connecting to an Essbase Server cluster, select the Cluster check box.**

Note: Essbase Server clusters must be set up prior to creating an Essbase connection. See *Oracle Hyperion Enterprise Performance Management System Installation and Configuration Guide* and *Oracle Hyperion Enterprise Performance Management System High Availability Guide* for information on setting up clusters.

8 **Enter the User Name and Password for this Essbase Server instance or cluster.**

9 **Select a Data Encryption method:**

- **No Encryption**—The default mode for communication between Essbase Studio Server and an Essbase Server instance or cluster.
- **SSL**—If the Essbase Server instance or cluster to which you are connecting is configured for Secure Socket Layer protocol over TCP/IP (SSL), select this method.

  SSL enables secure communication between Essbase Studio Server and Essbase.

  Note: If you select SSL, and Essbase Server is not configured to support SSL, then a connection cannot be established and an error message is displayed.

10 **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.

11 **Click Finish.**

You should see the connection name in the **Data 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 85.
- "Creating Metadata Elements for Performance Management Architect Sources" on page 86.

Defining Connection Parameters for Performance Management Architect Sources

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

1. In the **Data Source Navigator**, right-click **Data Sources** in the physical tree, select **New**, and then **Data Source.**
2. Enter a **Connection Name.**
3. **Optional:** Enter a **Description.**
4. In **Data Source Type**, select **Dimension Server.**

   Performance Management Architect is an example of a dimension server.
5. Under **Parameters**, in **Server Name**, enter the name of computer where Performance Management Architect Dimension Server resides.
6. **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.
7. Enter the **User Name** and **Password** for this Performance Management Architect Dimension Server instance.
8. Select a **Data Encryption** method:
   - **No Encryption**—This is the default mode for communication between Essbase Studio Server and a Performance Management Architect Dimension Server.
   - **SSL**—If the Performance Management Architect Dimension Server instance to which you are connecting is configured for Secure Socket Layer protocol over TCP/IP (SSL), select this method.

     SSL allows for secure communication between Essbase Studio Server and Performance Management Architect Dimension Server.

     Note: If you select SSL, and Performance Management Architect Dimension Server is not configured to support SSL, then connection cannot be established and an error message is displayed.
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 86.

If you clicked **Finish**, you should be able to expand the data source name in the **Data Source Navigator** to view the dimensions and members 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 **Data Source Navigator** to view the dimensions and members 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, review "About Text File Data Sources" on page 87, and then complete these procedures:

- "Defining Connection Parameters for Text File Sources" on page 87.
- "Modeling Text Files" on page 89.
- "Selecting a Minischema Option for Text File Sources" on page 90.
- "Creating Metadata Elements from Text File Sources" on page 92.

About Text File Data Sources

Text file data sources consist of flat text files that a database or system administrator places in a predefined location. By keeping the text file sources in one location starting from the indicated folder, administrators are able to limit user access to other files on the server, thereby providing a measure of security.

The default location for text files is:

EPM_ORACLE_HOME/products/Essbase/EssbaseStudio/Server/datafiles

Note: The server.datafile.dir property in the server.properties file is used to define the text file location. If you do not want to use the default text file location, see "server.datafile.dir" on page 34 for information on specifying the server.datafile.dir property.

Text files are contained within individual directories in the location specified by server.datafile.dir. Each text file directory is considered a text file data source, and can contain any number of individual text files.

Essbase Studio accesses the text file location at data source creation time. In the **Text File Location** dialog box, /<root> is the equivalent of the directory location defined by server.datafile.dir. The folders in this directory are the text file data sources from which the user can choose.

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 Data Source Navigator, right-click Data Sources in the physical tree, select New, and then Data Source.**

2 Enter a **Connection Name.**

3 **Optional:** Enter a **Description.**

4 Select **Text File** as the **Data Source Type.**

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

The location of the text file data source is specified using the `server.datafile.dir` property in the `server.properties` file. See "About Text File Data Sources" on page 87 for more information.

6 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 and individual text file names must not contain spaces.

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

8 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.**

9 **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**.

10 **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 (Col_0, Col_1, etc.) to each column from each file. You can change the column name later on a file-by-file basis when you model the text files, as described in "Modeling Text Files" on page 89.

11 Select a **Delimiter**:

- **Comma**
- **Tab**
- **Space**
- **Custom**—Specify a delimiter if your data source uses a delimiter other than 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.

12 **Click Next to proceed to the Model Text Files page of the wizard (described in** **“Modeling Text Files” on page 89****.**

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`,” you may 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 90.**

Selecting a Minischema Option for Text File Sources

You can create a minischema for a text file data source. Although only self-joins within a file are supported, a minischema for a text file data source can 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 contain only self joins (joins between columns in a single file). It will not contain joins between files.

  See "Creating a Minischema for Text File Sources" on page 91.

- Skip creation of a minischema diagram.

  See "Skipping Minischema Creation for Text File Sources" on page 91.

Note: The "Use Introspection to detect hierarchies" option is not supported for text files.

Creating a Minischema 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 91.

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 92

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.

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

   To add all tables to the minischema, click the "Add all tables to the schema" button.

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 only apply 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 92.

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, 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.

Synchronizing Data Source Connections

Perform these tasks for individual data source connections:

- "Performing an Incremental Update of Data Source Connections" on page 93
- "Deleting Tables from Data Source Connections" on page 96
- "Refreshing Tables in Data Source Connections" on page 97

Performing an Incremental Update of Data Source Connections

You may set up a data source connection to contain only a subset of the tables available in the physical data source. If you determine later that you want to add additional data source tables to the connection, you can perform an incremental update of the data source to add the 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 93
- "Performing an Incremental Update of a Text File Data Source" on page 94

Performing an Incremental Update of a Relational Data Source

This procedure applies only 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 94.

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 **Data Source Navigator**, 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.

2. Click 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 clicked "Show views" to display database views, click "Show views" again to stop displaying them.

3. In the **Available Tables** box, perform an action:

- Select the tables that you want to include in the data source connection, and click **Add selected tables to data source**.

  The tables that you selected are listed in the **Tables in Data Source** box.

- Click **Add all tables to data source** to add the tables to the **Tables in Data Source** box.

  All tables should appear in the **Tables in Data Source** box.

Note: When you select many tables to add to the connection, the incremental update process can take several minutes.

4 **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 newly added from the data source

  The tables that you just added are removed from the data source connection.

Note: You can remove only tables that you added during the current incremental update session. You cannot remove tables that are already included in the data source. Tables that are unavailable in the "Tables in Data Source" box cannot be removed.

5 **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*
```

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

Performing an Incremental Update of a Text File Data Source

This procedure applies only 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 93.

Note: This procedure applies only to text file data sources.

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

1 In the **Data Source Navigator**, 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.

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

4 **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, Essbase Studio assigns a default name to each column from each file.

5 **Select a Delimiter:**

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

6 **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:

- Text
- Integer
- Large integer
- Decimal

7 **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.

8 **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.

9 After you have finished selecting and modeling the text files that you are adding to the data source, click **OK** to close the **Incremental Update** dialog box.

Deleting Tables from Data Source Connections

Consider deleting tables that are no longer in use or that you do not want used with a particular data source connection.

You can delete tables from data source connections as long as no Essbase Studio metadata elements are dependent on the tables. For example, if a table is used to build a hierarchy stored in your Essbase Studio catalog, you cannot delete the table without first deleting the dependent hierarchy and its elements.

Note: You can only delete tables from relational data sources, including Oracle BI EE data sources, and files from flat file data sources.

➤ To delete tables from a data source connection:

1 In **Data Source Navigator**, under **Data Sources** in the physical tree, select a table belonging to a data source connection.

Use the **Shift** or **Ctrl** keys to select multiple tables.

Note: If you want to delete all tables from a data source connection, Oracle recommends deleting the data source connection (see "Deleting Connections" on page 105) and creating a new connection.

2 Right-click, and then select **Delete**.

3 If no tables are used by metadata elements or have joins to other tables, in the **Confirm Delete** dialog box, click **Yes**.

The **Confirm Delete** dialog box is displayed when no metadata elements depend on the selected tables.

4 If the selected tables are used by other metadata elements or have joins to other tables, perform these actions:

a. Review the metadata elements listed in the **Errors and Warnings** section of the **Delete** dialog box, and click **Cancel**.

Tip: Before clicking **Cancel**, click the **Save Errors and Warnings** button to save the list of tables as a text file, which can be used for reference in step 4.b.

b. Delete the noted metadata elements and all related metadata elements (see "Deleting Metadata Elements" on page 144).

Note: The Lineage Viewer can help you to quickly see dependent metadata elements, including Essbase models and cube deployments. Close the **Delete** dialog box, right-click a metadata element in the **Metadata Navigator**, and select **View Lineage**.

c. After deleting all dependent metadata elements, repeat the process by selecting a table to delete in the **Data Source Navigator**, and selecting **Delete**.

d. If the selected tables have joins with other tables, in the **Delete** dialog box, under **Table Status**, select the check box next to the table name to delete, and then click **Delete**.

If no other metadata element dependencies exist, the selected tables and joins listed for the selected table are deleted.

5 **Verify that the table is deleted in the Data Source Navigator.**

Refreshing Tables in Data Source Connections

Refresh tables to detect schema changes that occurred in your data source since you created the data source connection. Schema changes that are detected during refresh include:

- New columns
- Dropped columns
- Changes in column data type; for example, varchar to integer

You can refresh tables at the connection level and the table level. When you refresh at the table level, the refresh is performed only on the selected tables.

Note: You can refresh only relational data sources, including Oracle BI EE data sources.

The following topics describe the refresh process:

- "Refreshing Tables" on page 97
- "Adding Joins" on page 98

Refreshing Tables

Note: This procedure applies only to relational data sources, including Oracle BI EE data sources.

➤ To refresh data source connection tables:

1 **Perform an action:**

- Right-click one or more tables in a data source connection.

  Use the **Shift** or **Ctrl** keys to select multiple tables.
- Right-click a data source connection name.

2 **Select Refresh from the shortcut menu.**

Tables that are out of sync with the physical data source are flagged in the **Refresh** dialog box.

Note: If the selected tables or connection are already in sync with the physical data source, a message is displayed and no refresh action is performed.

3 **In the Refresh dialog box, perform these steps:**

a. Under **Changed Tables**, select a table.

b. Under **Changed Columns of Table**, review the information:

- **Column Name**
- **State**—Whether the column is new, updated, and so on
- **Data Type**
- **Change Details**

c. After reviewing the changes for each table listed under **Changed Tables**, select the tables that you want to refresh, and then click **Refresh**.

Note: Tables cannot be refreshed if outdated columns are still used in Essbase Studio metadata elements. You must remove such elements before you refresh the tables. The **Refresh** dialog box displays the tables that cannot be refreshed, along with the affected metadata elements.

d. If any joins will become invalid after the refresh, review them in the **Refresh** dialog box.

e. Select the tables that contain joins to be deleted, and click **OK**.

All joins associated with a table are deleted. You cannot select joins individually.

Alternatively, select the **Select All Joins** check box.

Note: You must explicitly select the tables containing the joins to be deleted. They are not deleted automatically during a refresh.

4 **Perform the steps in "Adding Joins" on page 98 to complete the process.**

Adding Joins

You can add joins after completing the steps in "Refreshing Tables" on page 97. After the data source is refreshed, Essbase Studio presents a list of potential joins for you to select.

Note: This procedure applies only to relational data sources, including Oracle BI EE data sources.

➤ To add new joins as part of the refresh process:

1 Perform the steps in "Refreshing Tables" on page 97.

After performing the refresh steps, the **New Joins** dialog box lists potential new joins.

2 In the **New Joins** dialog box, select the joins that you want to add as part of the data source connection refresh process.

3 Click **OK**.

A success message is displayed, showing updated tables.

Working with Data Source Connections

Perform these tasks in the **Data Source Navigator**:

Performing Introspection on a Data Source Connection

Introspection is a method of inspecting a physical data source for metadata elements. When you perform introspection, structural information 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.

➤ To launch introspection on an existing data source connection, in the **Data Source Navigator**, under **Data Sources** in the physical tree, right-click the data source on which you want to perform introspection and select **Introspect**.

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 the information you want to modify.

➤ To edit the properties of a data source:

1 **In the Data Source Navigator, under Data Sources in the physical tree, right-click the data source to edit and select Properties.**

2 **To change the data source connection name, enter a new name in Connection Name.**

3 **To change the data source description, enter a new description in Connection Description.**

4 **Complete the following tasks in the Parameters section of the dialog box as appropriate for the data source type.**

Relational sources:

Note: You cannot change the database name.

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

Note: 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.

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.

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.

f. Continue to step 5.

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. **Optional:** If you are connecting to an Essbase Server cluster, select the **Cluster** check box.

Note: Essbase Server clusters must be set up prior to creating an Essbase connection. See *Oracle Hyperion Enterprise Performance Management System Installation and Configuration Guide* and *Oracle Hyperion Enterprise Performance Management System High Availability Guide* for information on setting up clusters.

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

e. Select a **Data Encryption** method:

- **No Encryption**—The default mode for communication between Essbase Studio Server and an Essbase Server instance or cluster.
- **SSL**—If the Essbase Server instance or cluster to which you are connecting is configured for Secure Socket Layer protocol over TCP/IP (SSL), select this method.

  SSL enables secure communication between Essbase Studio Server and Essbase.

Note: If the Essbase Server is not configured to support SSL, then a connection cannot be established and an error message is displayed.

f. 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.

g. Continue to step 5.

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.

d. Continue to step 5.

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.

d. Continue to step 5.

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

6 If you changed the data source name and want to view the updated name in the **Data 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 Relational Source Tables and User-Defined Tables" on page 102.
- "Viewing Properties of Text File Source Files" on page 103.

Viewing Properties of Relational Source Tables and User-Defined Tables

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

1 In the **Data Source Navigator**, under **Data Sources** in the physical tree, expand the appropriate data source connection, and then navigate to the relational source table or user-defined table whose properties you want to view.

2 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 Relational Source Columns and User-Defined Table Columns" on page 102. Also see "Viewing the Statement on Which a User-Defined Table is Based" on page 103.

3 **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.

To view the properties of tables and columns in a minischema, see "Viewing Properties of Minischema Tables and Columns" on page 115.

Viewing Properties of 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 In the **Data Source Navigator**, under **Data Sources** in the physical tree, expand the appropriate data source connection, and navigate to the relational source column or the column in a user-defined table whose properties you want to view.

2 Right-click the column and select **Properties**, and then select the **Columns** tab.

These column properties are displayed:

- Whether a column is a key column
- Column name
- Column type
- Whether nulls are allowed

To view the properties of tables and columns in a minischema, see "Viewing Properties of Minischema Tables and Columns" on page 115.

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. **In the Data Source Navigator, under Data Sources in the physical tree, navigate to a user-defined table.**
2. **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. **In the Data Source Navigator, under Data Sources in the physical tree, navigate to a text file.**
2. **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 103.

   These properties are displayed:

   - Physical source file name
   - Data source connection name
   - Text file 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. **In the Data Source Navigator, under Data Sources in the physical tree, navigate to a text file source column.**
2. **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

In the **Data Source Navigator**, you can then view the available sample data for up to 200 rows of a table in a data source or from source columns within a table.

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

Viewing Sample Data from the Data Source Navigator

➤ To view sample data in a data source:

1 **In the Data Source Navigator, navigate to a data source and expand it.**

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 a 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, and then right-click and select **View Sample Data**.

The sample data 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.**

Note: If source data is added, modified, or deleted while the sample data viewing window is open, data does not automatically refresh in the open window. You must close the window and reissue the **View Sample Data** command to view the changed data.

Viewing Sample Data from the Metadata Navigator

➤ 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 element and select View Sample Data.**

If the caption binding and key binding expressions for the selected element are the same, only **View Sample Data** is displayed and there is no submenu.

If the caption binding and key binding expressions for the selected element are different, both the **With Caption Binding** option and the **With Key Binding** option are shown as submenu items of **View Sample Data**.

If the key binding expression for the element is **Delayed**, then the **With Key Binding** option is disabled.

The sample data that 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.

For example, using the TBC sample database, create a dimension element based on the REGION column of the REGION table where the caption binding expression equals `connection : \'tbcSource'::'tbc.region'.'REGION' . toString`; and key binding expression equals `connection : \'tbcSource'::'tbc.region'.'REGIONID' . toString`.

When you select the **With Caption Binding** option, the sample data results are:

```
    REGION
1   East
2   West
3   South
4   Central
```

When you select the **With Key Binding** option, the sample data results are:

```
    REGIONID
1   1
2   2
3   3
4   4
```

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

Refreshing the Connections List

➤ To refresh the list of data source connections in the **Data Source Navigator**, right-click **Data Sources** at the root of the physical tree 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 **Data Source Navigator**, right-click the name of the data source you want to delete, and select **Delete**.

Note: Connections that have been used to create metadata elements can be deleted only after the associated metadata elements have been deleted.

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 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 Microsoft SQL Server data source connection, in the **Data Source Navigator**, 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 Microsoft SQL Server data source connection, in the **Data Source Navigator**, right-click the data source connection name and select **Show Friendly Names**.

A check mark shows that **Show Friendly Names** is selected.

Working with the Data Source Connections of an Essbase Cube Deployed by Oracle BI EE

To take advantage of the aggregation power of Essbase, Oracle BI EE allows users to deploy Essbase cubes from the Oracle BI EE **Essbase Cube Generation** module.

In the **Essbase Cube Generation** module, users define measures, dimensions, and attributes, and then specify cube building options to deploy an Essbase cube. Essbase Studio serves as a web service to help Oracle BI EE deploy Essbase cubes. During the cube deployment process, Oracle BI EE sends an XML file which defines data source information and the cube schema with expected Essbase artifacts, along with deployment options to Essbase Studio. Essbase Studio then deploys the Essbase cube and returns the deployment results back to Oracle BI EE.

When an Essbase cube is deployed from Oracle BI EE, two elements are created in the **Data Source Navigator** in Essbase Studio Console:

- An Essbase connection element.

  This is the target Essbase Server connection to which the cube from Oracle BI EE is deployed. This object is prefixed with `Essbase` and the format is:

  `Essbase_<server>_<timestamp>`

  - `<server>`—the Essbase Server name specified in the Oracle BI EE XML (the server attribute of the `targetConnection` node).
  - `<timestamp>`—the generated timestamp for the operation in the time format:

    `MM-dd-yy_HH-mm-ss`

- An Oracle BI EE data source connection object.

  This is the data source connection of the Oracle BI EE Business Model from which the cube is defined. This node is prefixed with `OBIEE`.

  Since each Essbase cube deployment from Oracle BI EE must be differentiated in the **Data Source Navigator**, the naming convention for the nodes appear in this format:

  `OBIEE_<server>_<rpdFileName>_<timestamp[_source$srcNum]>`

  - `<server>`—the Oracle BI EE server address.
  - `<rpdFileName>`—the Oracle BI EE RPD file name.
  - `<timestamp_[_source<srcNum>]>`—the generated timestamp for the deployment in the time format:

    `MM-dd-yy_HH-mm-ss`

    For each redeployment, new Oracle BI EE data source connection nodes are created. Connection names are differentiated by appending `_source<srcNum>` to the connection string, and incrementing `<srcNum>` by `1` for each redeployment. If this is the first deployment, nothing is appended.

Work with Oracle BI EE data source connections created in this way as you would any Oracle BI EE data source connections that you create in Essbase Studio.

See also "Working with the Metadata Elements of an Essbase Cube Deployed by Oracle BI EE" on page 145.

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 "About User-Defined Tables" on page 108 and "User-Defined Table Examples" on page 108.

➤ To access the User-Defined Table Dialog Box:

1 **Highlight the name of the appropriate data source in the Data 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.

About User-Defined Tables

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, use user-defined tables 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.

See also "User-Defined Table Examples" on page 108.

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:

```
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`:

```
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`:

```
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
```

5 Minischemas

In This Chapter

About Minischemas

Minischemas are graphical models of the tables or text files in one or more data source connections. You can 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.

Minischemas enable you to bring in all tables or a subset of tables a data source connection in order to create a subject area on which to base further modeling.

You can perform these tasks in minischemas:

- Add or delete joins between tables, manually or by using inspection, where you select from a list of possible joins

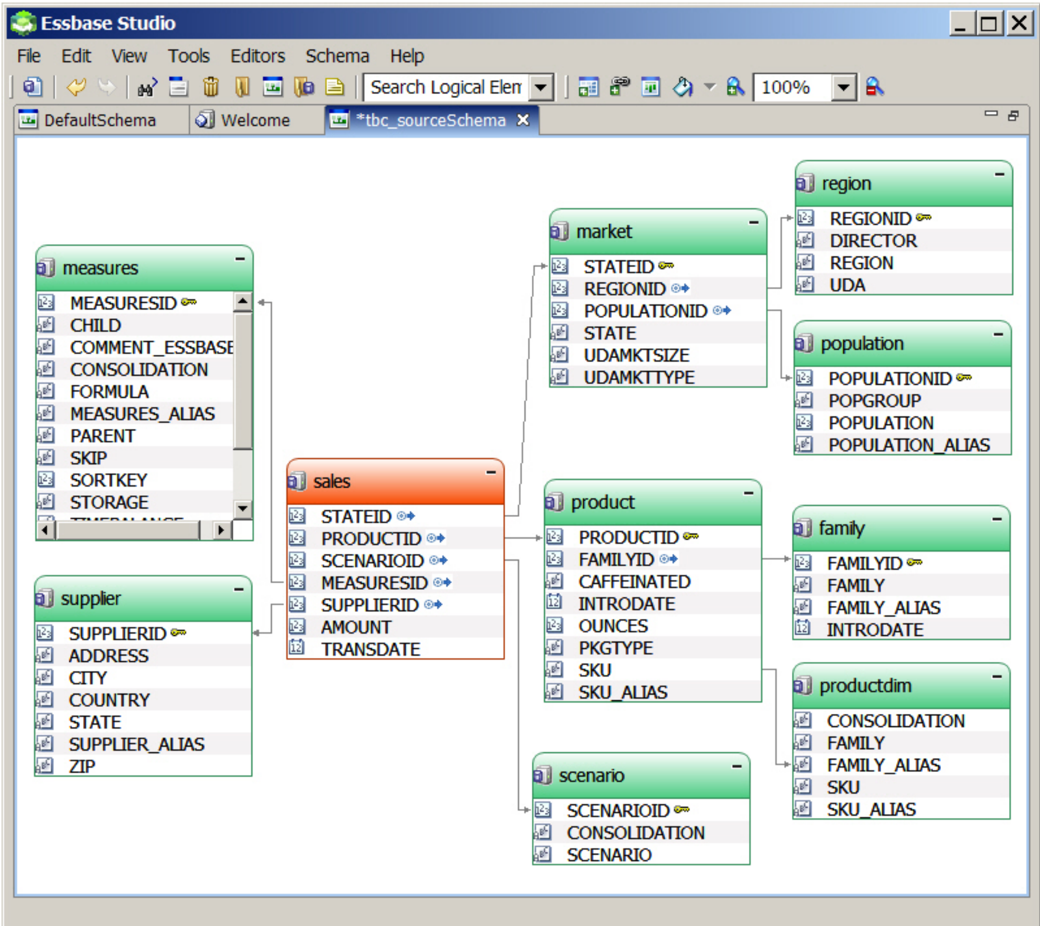
  Note: If you are accessing Oracle BI EE or text file data sources, you may create a minischema, but only self joins within a file are supported.

- View sample data from tables or columns
- Create metadata elements
- Add color to 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 79.

Figure 1 is a sample minischema.

Figure 1 Example Minischema



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

Creating or Editing Minischemas

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

- Name the new minischema, set a connection, and provide or edit a comment, as described in "Setting General Properties for Minischemas" on page 112.
- Add tables to or remove tables from the minischema and, optionally, add a filter,, as described in "Adding or Removing Tables in a Minischema" on page 113.

Setting General Properties for Minischemas

➤ To set the general minischema properties:

1 In the **Data Source Navigator**, perform an action:

- Navigate to the appropriate data source connection, right-click the **Minischemas** folder, and select **New**, and then **Minischema** to launch the **Minischema Wizard**.

  Note: You may also right-click on **Data Sources** at the top of the physical tree and select **New**, and then **Minischema**.

- For the appropriate data source connection, expand the **Minischemas** folder, right-click a minischema, and select **Properties**.

2 **For new minischemas, enter a Minischema Name.**

  If you are modifying a minischema, the **Minischema Name** field is not editable.

3 **For new minischemas, select a Connection from the drop-down list to associate with this minischema.**

  The **Databases** field displays the name of the database associated with this data source connection. This field is not editable.

  For existing minischemas, the **Connection** field is not editable.

  Note: If you began this procedure by right-clicking on **Data Sources** in the physical tree, then you must select the data source connection to associate with this minischema. If you began this procedure by right-clicking on a data source connection name, then the **Connection** field is already populated with that connection name.

4 **Optional: Enter or edit the Description.**

5 **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 113.

- 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 113.

  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 119.

Adding or Removing Tables in a Minischema

After naming the minischema and setting the connection, as described in "Setting General Properties for Minischemas" on page 112, 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/Remove Tables**.

➤ To add or remove tables from a minischema diagram:

1 **In the Add/Remove Tables page, note the contents of the Available Tables list, which is populated with all tables from the selected data source connection.**

2 From **Available Tables**, select the tables 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 the tables before moving them to the **Tables in Schema** list, enter the filter in the **Filter** text box.

4 **Optional:** For minischemas based on relational sources, to add related tables to the minischema, select tables in the **Tables in Schema** list, and click the **Add Related Tables** button.

You may want to add related tables if, when you created the data source connection, you did not select all available tables during the schema population procedure (see "Populating a Minischema" on page 73).

5 To remove tables from the minischema, in the **Tables in Schema** list, select the tables to remove and click the **Remove selected tables from the schema** button, which moves 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.

Removing tables from a minischema diagram does not remove or delete those tables from the data source connection on which the minischema is based.

Note: To remove a minischema table that has user-defined joins, Oracle recommends that you always delete those joins first, then delete the minischema table.

6 Click **Finish**.

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

8 **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, you can perform these tasks:

- "Saving a Minischema" on page 115
- "Opening a Minischema" on page 115
- "Removing Elements from a Minischema" on page 115
- "Adding or Editing Joins in a Minischema" on page 116
- "Adding Tables in a Minischema" on page 119
- "Viewing Sample Data" on page 120
- "Applying Color to Minischema Tables" on page 120

- "Creating Metadata Elements from Minischema Objects" on page 121
- "Viewing Minischemas" on page 121
- "Editing Properties of a Minischema" on page 125
- "Refreshing the Minischemas List" on page 126
- "Deleting Minischemas" on page 126

Saving a Minischema

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

Alternatively, select **File**, and then **Save**; or press **Ctrl+S**.

Opening a Minischema

➤ To open a minischema for editing:

1. In the **Data Source Navigator**, expand the appropriate data source connection, and then expand the **Minischema** folder.
2. Right-click the minischema name in the tree, and select **Edit**.

Viewing Properties of Minischema Tables and Columns

➤ To view properties of minischema tables and columns:

1. If not already opened, open the minischema (see "Opening a Minischema" on page 115).
2. In the minischema work area, right-click the table object whose properties you want to view and select **Properties**.

   The table properties are displayed in the **General** tab of the properties dialog box.
3. Click the **Columns** tab to view these column properties:
   - Whether a column is a key column
   - Column name
   - Column type
   - Whether nulls are allowed

Removing Elements from a Minischema

You can modify a minischema by removing tables or joins from it.

You can follow the dialog box-based procedure described in "Adding or Removing Tables in a Minischema" on page 113 to remove tables from a minischema. Or, you can use the process described below to remove tables or selected joins.

Note: To remove a minischema table that has user-defined joins, Oracle recommends that you always delete those joins first, then delete the minischema table.

➤ To remove tables or joins from a minischema:

1. **If not already opened, open the minischema (see "Opening a Minischema" on page 115).**
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 inspect your data source to determine possible logical joins for you.

- To add joins manually, see "Adding or Editing Joins Manually" on page 116.
- To add joins by inspection, see "Adding Joins By Inspection" on page 118.

Adding or Editing Joins Manually

Essbase Studio uses the same dialog box to manually create and edit joins.

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

You can edit joins by changing the columns used in a join pair, adding more join pairs, or modifying other criteria.

➤ To add or edit a join:

1. **If not already opened, open the minischema (see "Opening a Minischema" on page 115).**
2. **Perform a task:**
   - 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 of the **Edit Properties of Minischema Join** dialog box.

Note: Because you can only create self joins in minischemas created from Oracle BI EE and 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 **Table** drop-down lists. You select the tables you want to join.

Note: For self joins in minischemas created from Oracle BI EE or text file data sources, only the left **Table** drop-down list box is active. Once you select a table in the left Table drop-down list box, the right **Table** text box also displays that table name.

- To edit a join, double-click a 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, select the table to which you want to establish a join.**

Note: For minischemas built from Oracle BI EE or 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; for example, joins from relational data sources:**

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 an option:

- **Left**—Returns all rows from the left-hand table and any matching rows from the right-hand table.
- **Full**—Returns all matching and non-matching rows from both the left-hand and right-hand tables.
- **Right**—Returns all rows from the right-hand table and any matching rows from the left-hand table.

If you do not select the **Outer** check box, then an "inner" join is created by default. With inner joins, only matching rows from both tables are returned.

f. **Optional:** Repeat step 2 through step 5.e for each join combination that 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; for example, self joins from relational, Oracle BI EE, or text file data sources:**

a. In the left **Column** drop-down list, select the column name to being 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

Join options are described in step 5.e.

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.

7 **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 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 inspect your database and return a list of possible join pairs.

➤ To add joins by inspection:

1 **If not already opened, open the minischema (see "Opening a Minischema" on page 115).**

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 a task:**

- 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 a task 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.**

6 **Verify that the selected join pairs were added in your minischema diagram.**

7 **To save 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 in "Adding or Removing Tables in a Minischema" on page 113 or use the process below.

➤ To add tables to a minischema:

1 **If not already opened, open the minischema (see "Opening a Minischema" on page 115).**

2 **In the Data Source Navigator,under the data source connection name, select the table you want to add and drag it into the minischema work area.**

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

Note: You cannot add the same physical table to the minischema more than once.

3 **Repeat step 2 for all physical tables that you want to add to the minischema.**

4 **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 individual columns.

➤ To view sample data in a minischema:

1 **If not already opened, open the minischema (see "Opening a Minischema" on page 115).**

2 **Perform either or both of the following tasks:**

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

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

Note: If source data is added, modified, or deleted while the sample data viewing window is open, data does not automatically refresh in the open window. You must close the window, reselect tables or columns, and reissue the **View Sample Data** command to view the changed data.

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 large minischema with multiple fact tables. You can set the fact tables to red or another color to more easily to locate them 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. See Figure 1 on page 112 for an example of color in a minischema.

➤ To apply color to minischema table:

1 **If not already opened, open the minischema (see "Opening a Minischema" on page 115).**

2 **Select a table element to which to add color.**

To select multiple tables, press and hold the **Ctrl** key, then click on the tables to which you want to add color.

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

3 **Right-click and select Color to view available colors.**

4 **Select a color from the list.**

5 **View the effect on the minischema diagram in the minischema work area.**

6 **To save your changes, right-click in the minischema view and select Save Minischema.**

Creating Metadata Elements from Minischema Objects

You can create metadata elements from objects in a minischema quickly and easily 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 115****).**
2. **Select 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 metadata elements from multiple tables in the minischema, press and hold the **Shift** key, then click on tables to select them, right-click, and then select **Add to Metadata Navigator**.
   - To create a metadata element from one column in a minischema table, select the column, right-click, and then select **Add to Metadata Navigator**.
   - To create metadata elements from multiple columns in one or more minischema tables, press and hold the **Ctrl** key down, then click the columns in the minischema tables to select them, right-click, and then select **Add to Metadata Navigator**.

   After performing an action 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. in the “**Choose Folder**” dialog box, click the **Create Folder** button 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 4 for all metadata elements that you want to create.**

Note: You may also use drag-and-drop to create metadata elements. Select the elements from the minischema as described in step 2, but do not right-click. Instead, drag the selected elements to the appropriate folder in the **Metadata Navigator**. Be aware that when using this method, you will not have the option to create folders during the process.

Viewing Minischemas

Because some minischema diagrams can be very large, 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 122.

- 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 124.

- Use the zoom 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 124.

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 the minimized and maximized views of the minischema in the work area (see “Maximizing and Minimizing the Minischema Work Area” on page 124).

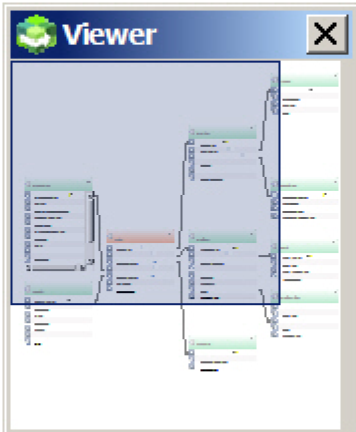
➤ To use the thumbnail viewer:

1. **If not already opened, open the minischema (see “Opening a Minischema” on page 115).**

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

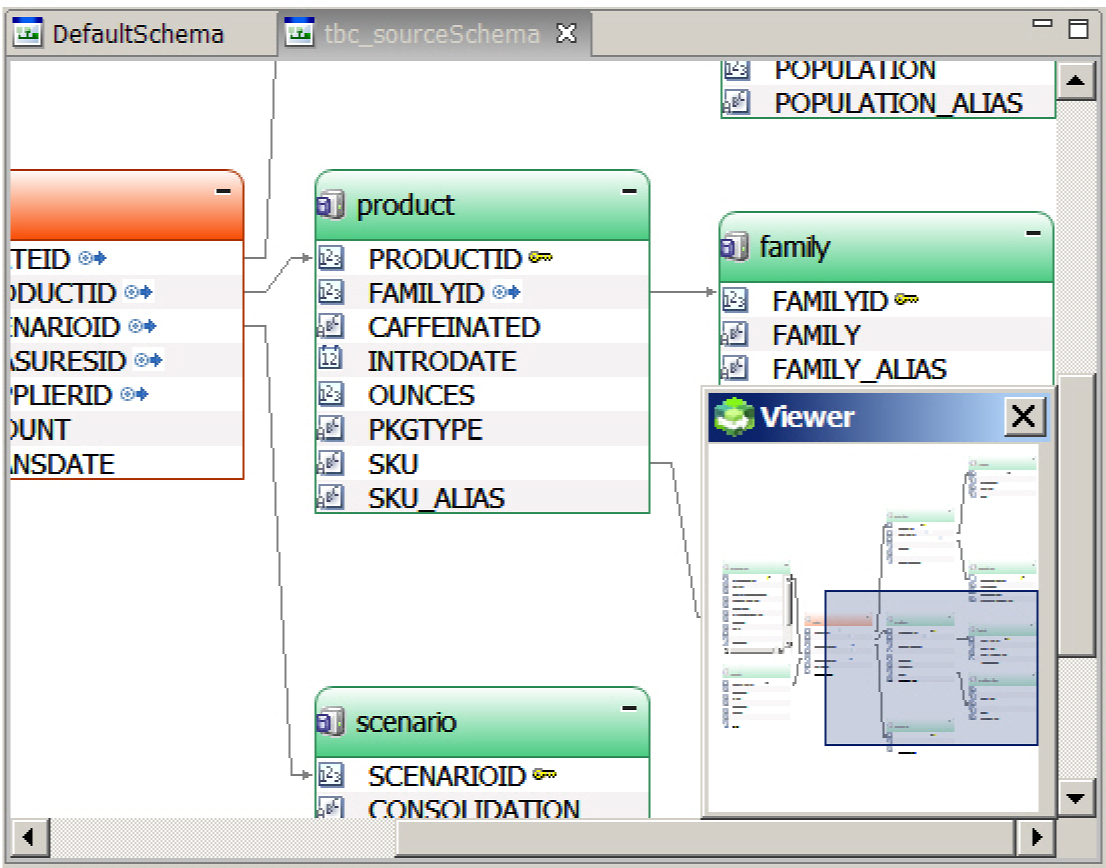
   The thumbnail viewer pops up in the lower right corner of the minischema work area. The thumbnail viewer contains the same portion of the minischema as displayed in the main minischema work area, but in a miniature format. A smaller, transparent blue pointer covers a portion of the minischema.

Figure 2 Thumbnail Viewer



3 **Drag the pointer in any direction to the location in the minischema work area that you want to view.**

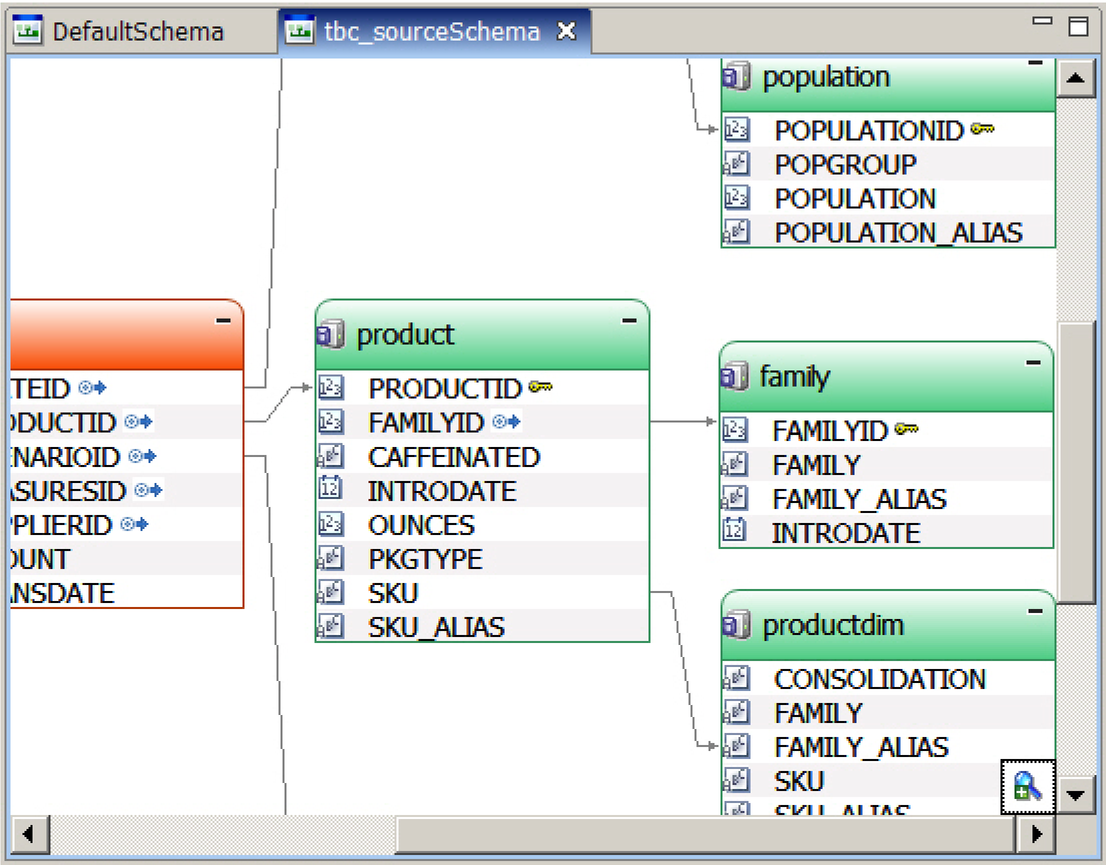
Figure 3 The Thumbnail Viewer and the Point of Focus on the Minischema Work Area



4 **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.

Figure 4   The Focused Portion of the Minischema Work Area After Closing 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 **Data Source Navigator**. You can expand the minischema work area to help you to navigate 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 122).

➤ 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 minischema tables wherever you want in the minischema work area, or Essbase Studio can arrange the tables for you.

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

Note: You can also automatically arrange the tables by right-clicking in the minischema work area and selecting **Lay out schema**.

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.

You can perform these editing tasks:

- 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 115).
2. In the **Data Source Navigator**, expand **Minischemas** in the physical tree, right-click a minischema to edit, and select **Properties.**
3. Follow the instructions for editing minischema properties in “Adding or Removing Tables in a Minischema” on page 113.

   Note: The instructions for editing minischema properties are the same procedures documented in “Adding or Removing Tables in a Minischema” on page 113.

Refreshing the Minischemas List

➤ To refresh the list of minischemas in the **Minischemas** folder of a data source connection:

1. In the **Data Source Navigator**, expand a data source connection.
2. Right-click the **Minischemas** folder and select **Refresh.**

Deleting Minischemas

➤ To delete a minischema>, .

1. In the **Data Source Navigator**, expand a data source connection, and expand the **Minischemas** folder.
2. Right-click the name of the minischema you want to delete and select **Delete.**

Deleting a minischema does not delete the data source connection on which the minischema is based.

Note: When you want to delete a minischema that has user-defined joins, Oracle recommends that you always delete those joins first, then delete the minischema.

6

Introspection

In This Chapter

Introspection Overview

Introspection is a method of inspecting a physical data source for metadata elements. When you perform introspection, structural information that 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" it for metadata elements.

Note: Introspection is not supported for Oracle BI EE or text file data sources.

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

- Fact tables
- Dimension tables
- Hierarchies

During the introspection process, you also can create a minischema.

The Introspection Wizard guides you through the process. The wizard allows you view the candidate selections and choose which items to keep. These topics describe the process:

- "Selecting a Minischema Option in the Introspection Wizard" on page 128.
- "Selecting Fact Tables in the Introspection Wizard" on page 128.
- "Selecting Dimension Tables in the Introspection Wizard" on page 129.
- "Selecting Hierarchies in the Introspection Wizard" on page 129.

Selecting a Minischema Option in the Introspection Wizard

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

➤ To begin introspection and choose a minischema option:

1 **In the Data 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.
- **Skip minischema diagram**—If you choose this option, you still are 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 a 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 80.

Selecting Fact Tables in the Introspection Wizard

During introspection, 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 contains 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 one or more tables from **Available Tables** to use as fact tables.

Note: The **Available Tables** list contains all the remaining tables, views, alias tables, and synonyms that were selected when the data source was created.

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 that pertain to area, such as:

```
perf*
```

Filters apply to all tables, views, aliases, or synonyms you have included in the selected data source.

3 **When you have made your selections, click Next to view the Select Dimension Table(s) page of the Introspection Wizard.**

Selecting Dimension Tables in the Introspection Wizard

During introspection, 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 that you want to designate as dimension tables.**

The **Selected Dimension Table(s)** list may already contain 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 objects from **Available Objects** to designate 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 that pertain to that area, such as:

```
perf*
```

Filters apply to all tables, views, aliases, or synonyms you have included in the selected data source.

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 Hierarchies list.**

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 multichain 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 **Create 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, in the **Available Tables and Columns** list, navigate to the first column to add to the hierarchy, select it, and click the **Add column as child** button.

     The column you added is now a member in the hierarchy

     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.

     Note: Be sure to first select either the hierarchy name or the correct member in the hierarchy before selecting columns to add as child or sibling members.

- 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: `*tablename*Hierarchy`.

  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.

     Note: Be sure to first select either the hierarchy name or the correct member in the hierarchy before selecting columns to add as child or sibling members.

2 **Click Finish.**

If you created a minischema, you can review it in the Minischema work area.

Also, review the metadata elements created by the introspection process. They are listed in the **Metadata Navigator** under the appropriate folder.

7 Metadata Elements

In This Chapter

About Metadata Elements

Metadata elements are the logical objects derived from the physical objects in a data source. Metadata elements can be created when you create a data source connection or afterward. 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, with these features:
  - Created during the data source connection process or at any time after.
  - Text-, numeric-, or date-based.
  - Editable to specify bindings other than the physical column on which they are based.
  - Editable to add a filter.
  - Editable to add a sort order.
- Derived text measures—A text measure whose values are governed by a predefined rule expressed as a range.
- Text lists—Pre-mapped values and IDs that can be associated with Essbase members.
- 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 (described in Chapter 9, "Hierarchies").

A folder is another type of metadata element. Folders help organize items in the **Metadata Navigator**. See "Creating or Editing Metadata Folders" on page 143.

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 132
- "Creating or Editing Derived Text Measures" on page 139
- "Creating or Editing Text Lists" on page 142
- "Creating Date Elements" on page 142
- "Creating or Editing Metadata Folders" on page 143

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 132.
  - 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 134.
- 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 139.
- 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 141.

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 Source Navigator
- 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** and then **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 134 to define the binding expression and, optionally, filters and sort order.

Creating a Dimension Element from the Data Source Navigator

➤ To create a dimension element from the **Data Source Navigator**:

1 In the **Data Source Navigator**, navigate to the physical element upon which you want to base the dimension element.

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

Note: You can select multiple table or column elements in the **Data Source Navigator**.

3 **Optional:** Proceed to "Defining or Editing General Properties for Dimension Elements" on page 134 to edit the binding expression and define filters and sort order.

Creating a Dimension Element from a Minischema

➤ To create a dimension element from the minischema:

1 In the **Data Source Navigator**, expand the data source connection, expand the **Minischemas** folder, and select the minischema from which you will create dimension elements.

2 Right-click the minischema and select **Edit.**

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

3 In the minischema work area, drag an element from the diagram (such as an entire table or an individual column), and drop it 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, right-clicking, and selecting **Add to Metadata Navigator**.

4 **Optional:** Proceed to "Defining or Editing General Properties for Dimension Elements" on page 134 to edit the binding expression and define filters and sort order.

Defining or Editing General Properties for Dimension Elements

Before you define or edit the general properties for dimension elements, create a dimension element using a methods described in "Creating or Editing Metadata Elements" on page 132.

Derived text measures must be created and edited using the method described in "Creating or Editing Derived Text Measures" on page 139.

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 an action:**

- To create a dimension element, complete a procedure described in "Creating Dimension Elements" on page 132.
- To edit a dimension element, locate it in the **Metadata Navigator**, right-click, and select **Edit**.

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

2 **Create an expression on which to base the dimension element.**

Follow the instructions in "Creating an Expression on Which to Base a Dimension Element" on page 135.

3 **Choose a key binding option.**

Follow the instructions in "Choosing a Key Binding Option for a Dimension Element" on page 137.

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 "Creating an Expression on Which to Base a Dimension Element" on page 135.

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 a column in the dimension, drag the column from the Source tab in the Formula list and drop it under Sort Column; then, in the Sort Order column, click the down arrow 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 of the 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 141.**

Creating an Expression on Which to Base a Dimension Element

➤ To create an expression:

1 **If you have not already done so, create a dimension element or open one for editing.**

- To create a dimension element, complete a procedure described in "Creating Dimension Elements" on page 132.
- To edit a dimension element, locate it in the **Metadata Navigator**, right-click, and select **Edit**.

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

2 **In the Edit Properties dialog box, create an expression on which to base the dimension element.**

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.

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 "Rules for Entering User-Defined Expressions" on page 136.

Notes:

- By default, dimension elements, whether they are text-, numeric- or date-based, retain the same datatype as the physical element on which they are based. For example, if you create a dimension element based on AMOUNT from the SALES table in the TBC database, the datatype, or unit, of that dimension element is now numeric. The key binding expression for that element must always be numeric.

  You can use operands with different unit types in the key binding expressions but the expression result should return the values with the unit that is compatible with the unit of the dimension element. If the unit of the dimension element is a string, you can apply the property `.toString`to any non-string expression in order to convert the expression result to string.

  For example, suppose you created a dimension element based on the FAMILY column from the FAMILY table in the TBC database. The caption and key binding expression for this dimension element is:

  ```
  connection : \'tbcSource'::'TBC.tbc.FAMILY'.'FAMILY'
  ```

You can change the key binding expression to a numeric-based expression by appending `.toString` as follows:

```
connection : \'tbcSource'::'TBC.tbc.FAMILY'.'FAMILYID'.toString
```

- 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".
- For caption binding and key binding expressions, text file data sources only support the substring (substr) function and the concatenation operator (||) .
- 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 356 for more information.
- For limitations on creating expressions for independent dimensions, see "Independent Dimension Bindings Limitations" on page 362.

Use the following steps as guidelines 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.

Rules for 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 137.
- "Entering Strings in User-Defined Expressions" on page 137

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 `ab` must be entered as `"ab"`

  The string `a'b` must be entered as `"a'b"`

- Double quotes within a string must be repeated; for example:

  The name `a"b` must be entered as `"a""b"`

- Double quotes preceding or following a string must be preceded and followed by two double quotes; for example:

  The name `"a"b"` must be entered as `"""a""b"""`

Choosing a Key Binding Option for a Dimension Element

➤ To choose a key binding option:

1 **If you have not already done so, create a dimension element or open one for editing.**

- To create a dimension element, complete a procedure described in "Creating Dimension Elements" on page 132.
- To edit a dimension element, locate it in the **Metadata Navigator**, right-click, and select **Edit**.

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

2 **If you have not already done so, complete the procedure in "Creating an Expression on Which to Base a Dimension Element" on page 135.**

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 "Creating an Expression on Which to Base a Dimension Element" on page 135 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 To complete the dimension element creation or editing process, return to step 4 on page 134 in "Defining or Editing General Properties for Dimension Elements" on page 134.

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 be 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 141.

Note: When derived text measures are used in cube schemas to build Essbase models, the "XOLAP Model" option is unavailable 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, then 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.

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 356 for more information.

4 **In Range, click the plus sign (+) sign and enter ranges and their corresponding text value.**

Click the plus sign (**+**) to enter each new range on the next row.

The text values you enter in the **Label** column will appear in a report if this derived text measure is used in a cube schema that is later deployed as an Essbase cube.

You can enter precise numeric range values, as shown in Table 1, or use the range keywords, as shown in Table 2.

Following is an example set of ranges for sales figures:

Table 1 Ranges Using Precise Numeric Values

From Value	To Value	Label
0	25000	Poor
25001	75000	Good
75001	100000	Excellent

Table 2 Ranges Using Range Keywords

From Value	To Value	Label
below	25000	Poor

From Value	To Value	Label
25000	75000	Good
75000	above	Excellent

To use range keywords:

a. In the **From Value** column, click in the cell to activate the drop-down and select **<below>** from the list.

b. In the **To Value** column, click in the cell to activate the drop-down and select **<above>** from the list.

5 **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 141.

Creating or Editing Alias Set Bindings for a Given Alias Set

You can create alias set bindings using the Alias Set dialog box. 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 reflected in the bindings for the selected alias set.

➤ 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 an action:

- To create a dimension element in the **Metadata Navigator**, right-click the folder where you want to store the dimension element, and select **New**, then **Dimension Element**.
- To edit a dimension element, locate it in the **Metadata Navigator**, right-click, and select **Edit**.

The **Edit Properties** dialog box is displayed.

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 that you want to bind to this element, and click **Apply**.

5 **Optional:** To delete an alias binding, select a row in the **Alias Sets** grid and click **Delete.**

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

7 When finished, click **OK.**

Creating or Editing Text Lists

Text lists are metadata elements that map a column containing text strings to a column containing IDs for those strings. The columns you use in the text list are columns from your data source, usually from a specific data source table that contain the text strings and IDs. You set up this structure before you create text lists in Essbase Studio.

Text lists are used in conjunction with text measures. Text measures extend the analytical capabilities of Essbase beyond numerical data to text-based content. Storage and analysis of textual content can be useful when a cell needs to have one of a finite list of textual values.

For example, a product may be sold in five different colors. The color is a text measure whose value must be one of the five colors. The colors are a set of text strings mapped to corresponding numeric IDs. These mappings are contained in the text list element that you create.

➤ Before you create or edit a text list:

1. Ensure that a database table exists with columns containing the IDs and values you plan to use in your text list.
2. Ensure that the database columns you plan to use in your text list exist as dimension elements in the **Metadata Navigator**.

➤ To create or edit a text list:

1. Perform an action:
   - To create a new text list, in the **Metadata Navigator**, right-click on the folder where you want to store the text list and select **New**, then **Text List**.
   - To edit a text list, locate it in the **Metadata Navigator**, right-click it, and select **Edit**.
2. In the **Properties** dialog box, enter a **Name** and optional **Description** for this text list.
3. Under **Bindings**, select the column from the data source that contains the data values, and click the right arrow button to move it to **Value Bindings**.
4. **Optional:** Under **Bindings**, select the column from the data source that contains the numeric values, and click the right arrow button to move it to **ID Binding**.
5. Click **OK** to save the text list.

   Note: If this is a new text list, verify that it appears in the **Metadata Navigator**.

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 check box next to the date elements that you want to create.**

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

If you chose to create metadata elements at data source creation time, the Connection Wizard gives you the option of creating a metadata folder to store those elements. However, you can create metadata folders at any time and use them to store not only metadata elements, such as dimension elements or hierarchies, but other objects, such as cube schemas or drill-through reports. Metadata folders can help you organize the items in the **Metadata Navigator**.

➤ To create a metadata folder:

1 In the **Metadata Navigator**, navigate to the location where you want to create the folder, and select **File**, then **New**, and then **Folder.**

2 Enter a folder **Name.**

3 **Optional:** Enter a **Description.**

4 Click **Apply**, then click **OK.**

➤ To edit a folder name or description:

1 Right-click the folder name and select **Edit.**

2 **Optional:** Update the **Name.**

3 **Optional:** Update the **Description.**

4 Click **Apply**, then click **OK.**

Working with Metadata Elements

You can perform these tasks with metadata elements:

- "Copying Metadata Elements" on page 144
- "Renaming Metadata Elements" on page 144
- "Deleting Metadata Elements" on page 144
- "Viewing Sample Data for Metadata Elements" on page 145
- "Showing Lineage" on page 145
- "Working with the Metadata Elements of an Essbase Cube Deployed by Oracle BI EE" on page 145

Copying Metadata Elements

➤ 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 an action in the **Metadata Navigator**:

- Double-click the element to launch the element's properties dialog box and enter the new name in the name field.
- Right-click the element and select **Rename**, then enter 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**.

You cannot delete metadata elements if other metadata elements are dependent on them. You must first delete the dependent metadata elements. For example, you cannot delete a hierarchy that is used in a cube schema and as a drill-through intersection until you delete the affected drill-through reports, Essbase models, cube schemas, etc.

If an error message box is displayed when trying to delete a metadata element, click the Details button to determine the item that is dependent on the metadata element that you are trying to delete. This is an iterative process; the error message box displays each time you attempt to delete the metadata element until all dependent metadata elements are first deleted.

Note: The Lineage Viewer can help you to quickly see dependent metadata elements. Right-click the metadata element in the **Metadata Navigator** and select View Lineage.

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**.

See "Viewing Sample Data" on page 104 for more information on viewing sample data from metadata elements in the **Metadata Navigator**.

Showing Lineage

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

For information on lineage, see Chapter 15, "Lineage."

Working with the Metadata Elements of an Essbase Cube Deployed by Oracle BI EE

To take advantage of the aggregation power of Essbase, Oracle BI EE allows users to deploy Essbase cubes from the Oracle BI EE **Essbase Cube Generation** module.

The Oracle BI EE interface may be used to define and deploy Essbase cubes. In Oracle BI EE, users define measures, dimensions, and attributes, and then specify cube building options to deploy an Essbase cube. Essbase Studio serves as a web service to help Oracle BI EE deploy Essbase cubes. During the cube deployment process, Oracle BI EE sends an XML file which defines data source information and the cube schema with expected Essbase artifacts, along with deployment options to Essbase Studio. Essbase Studio then deploys the Essbase cube and returns the deployment results back to Oracle BI EE.

When an Essbase cube is deployed from Oracle BI EE, the metadata elements derived from the Oracle BI EE data source are stored in the **Metadata Navigator** of the Essbase Studio Console.

When the first deployment from Oracle BI EE occurs, a folder is automatically created under `Root` in the **Metadata Navigator** in the format:

```
OBIEE_<server>_<rpdFileName>_<timestamp>
```

- `<server>`—the machine name of the Essbase Server.
- `<rpdFileName>`—the Oracle BI EE RPD file name.
- `<timestamp>`—the generated timestamp for the operation in the time format:

  ```
  MM-dd-yy_HH-mm-ss
  ```

For each redeployment, a new folder is created, differentiated by the `<timestamp>` in the **Metadata Navigator**.

The folder contains the dimension elements, hierarchies, cube schema, Essbase model, and Essbase application and database that were created during the deployment.

Work with the metadata elements in the folder as you would with any other metadata elements that you create from other Oracle BI EE Business Model data sources. For example, customize and optimize elements to create new cube schemas and Essbase models, then deploy those models to create new Essbase cubes.

See also "Working with the Data Source Connections of an Essbase Cube Deployed by Oracle BI EE" on page 106.

8 Alias Sets

In This Chapter

About Alias Sets

The use of 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, stored in alias tables, 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 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 shows some products in the Sample Basic database that have two descriptive alias names:

Product	Default	Long Names
100-10	Cola	The Best Cola
100-20	Diet Cola	Diet Cola with Honey
100-30	Caffeine Free	Cola All the Cola, none of the Caffeine

An alias set maps a specific, named set of alias names to member names. Unlike Essbase, there is no default alias table. You create and store alias sets in any folder, including the root folder, in the **Metadata Navigator**. Users can edit, copy, rename, delete, and export alias sets on the **Metadata Navigator** by using the right-click menus. 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 147.

Working with Alias Sets

From the **Metadata Navigator**, you can perform the following tasks:

- Create alias sets—See "Creating Alias Sets" on page 148.
- Modify, add, or delete bindings in alias sets—See "Editing Alias Sets" on page 150.
- Copy, rename, delete, or export alias sets—See "Managing Alias Sets" on page 152.

For an overview of alias sets, see "About Alias Sets" on page 147.

Creating Alias Sets

Alias sets are the container that holds the alias bindings that you create.

➤ To create an alias set:

1. **In Metadata Navigator, right-click on the folder where you want to store the alias set and select New, then Alias Set.**

   The **Alias Set** dialog box is displayed.

2. **Enter a Name and an optional Description for the alias set.**
3. **Perform either of the following tasks to create one or more bindings for this alias set.**
   - "Creating Bindings Manually for an Alias Set " on page 148
   - "Creating Bindings By Inspection for an Alias Set" on page 149
4. **After adding bindings to the alias, set, click OK and view the new alias set listed in the Metadata Navigator.**

Creating Bindings Manually for an Alias Set

After you have created an alias set (as described in "Creating Alias Sets" on page 148), you can add bindings to it. Bindings specify the columns in the data source that hold alias information for a particular alias set.

In this procedure, you create the bindings manually. You can also create bindings automatically, by inspection. For instructions, see "Creating Bindings By Inspection for an Alias Set" on page 149.

➤ To create bindings manually for an alias set:

1. **If have not already done so, in the Metadata Navigator, right-click the alias set to which you want to add bindings and select Edit to launch the Alias Set dialog box.**

   Note: Before you can add bindings to an alias, you must have created an alias set, as described in "Creating Alias Sets" on page 148. If you have already created an alias set, proceed to step 2.

2. **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 includes 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.

3 **In the Binding source tree, select a metadata element for which you want to associate an alias binding.**

4 **In the source tree under the Binding expression text box, navigate to the column that you want to use to generate a binding expression.**

5 **Double-click the column name to add the binding expression to the text column.**

Use the source tree below the **Binding expression** text box to generate a binding expression for the metadata object that you selected in the **Binding source** tree. Alternatively, you can enter an expression in this text box.

6 **Click OK.**

The expression you created is displayed in the **Bindings** section of the **Alias Set** dialog box in the following format:

- Under **Dimension element**, the metadata element that you are associating with an alias is displayed.
- Under **Binding**, the source element on which you are basing the alias is displayed.

7 **Repeat step 2 through step 6 for all bindings that you want to create for this alias set.**

8 **When you have created all bindings for an alias set, click OK to close the Alias Set dialog box.**

Creating Bindings By Inspection for an Alias Set

After you create an alias set (as described in "Creating Alias Sets" on page 148), you can add bindings to it. Bindings specify the columns in the data source that hold alias information for an alias set.

In this procedure, you create the bindings by inspection. You can also create bindings manually. See "Creating Bindings Manually for an Alias Set " on page 148.

➤ To create bindings by inspection for an alias set:

1 **If have not already done so, in the Metadata Navigator, right-click the alias set to which you want to add bindings and select Edit to launch the Alias Set dialog box.**

Note: Before you can add bindings to an alias, you must have created an alias set, as described in "Creating Alias Sets" on page 148. If you have already created an alias set, proceed to step 2.

2 **To the right of the Bindings box, click Create bindings by inspection.**

3 **In Create Aliases, locate Pattern for alias columns, and enter a string that represents all or part of a column name that you will use to base bindings.**

For example, if each table in a database contains a column named in the format, `<language>_alias`, enter this search pattern:

```
*alias
```

4 **Select data source connections that you want to inspect for alias bindings.**

5 **Click OK.**

Essbase Studio inspects the selected data source connections for alias bindings and lists them under **Bindings** in the **Alias Set** dialog box.

6 **Review the alias set bindings presented in the Alias Set dialog box, editing or deleting them as necessary.**

7 **When reviewing the bindings created for the alias set, click OK to close the Alias Set dialog box.**

Editing Alias Sets

You can perform these editing functions on the bindings in alias sets:

- Sort bindings in ascending or descending order by dimension element name, dimension element path, or binding string—See "Sorting Dimension Elements and Bindings in an Alias Set" on page 150.
- Modify an existing binding in the set—See "Modifying Bindings in an Alias Set" on page 151.
- Add bindings to the set—See "Adding Bindings to an Alias Set" on page 151.
- Delete bindings from the set—See "Deleting Bindings from an Alias Set" on page 152.

Sorting Dimension Elements and Bindings in an Alias Set

You can sort the alias set bindings in ascending or descending order by the dimension element name, dimension element path, or binding string. Sorting eases locating the binding that you want to work with.

➤ To sort dimension elements and bindings:

1 **If have not already done so, in the Metadata Navigator, right-click the alias set to which you want to apply sorting and select Edit to launch the Alias Set dialog box.**

2 **Under Bindings, select a binding, and then right-click.**

3 **From the popup menu, select an option:**

- **Sort by dimension element name**
- **Sort by dimension element path**

  Note: To use this option, select the **Show full path for dimension elements** option to enable it. A check mark appears next to this option when it is enabled. When selected, the full path and dimension element name appear in the **Dimension element** column.

- **Sort by bindings**
- **Sort ascending**

- **Sort descending**
- **Show full path for dimension elements**

4 **Optional:** After you locate the binding to work with, you can continue with one of these tasks:

- "Modifying Bindings in an Alias Set" on page 151
- "Deleting Bindings from an Alias Set" on page 152

Modifying Bindings in an Alias Set

You can modify bindings in an alias set.

➤ To modify a binding:

1. If have not already done so, in the **Metadata Navigator**, right-click the alias set in which you want to modify bindings and select **Edit** to launch the **Alias Set** dialog box.
2. Under **Bindings**, select the binding to modify.
3. Click the **Edit the selected binding** button to the right of **Bindings** box to display the **Edit a Binding** dialog box.

   The top pane displays the current binding syntax.
4. Perform an action:
   - Enter the binding expression into the top pane of the dialog box; the syntax is:

     ```
     connection :
     \'<logical_source_name'::'<physical_data_source_name>.<physical_t
     able_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 that 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, Oracle recommends that you clear the text from the top pane of the dialog box before you begin.
5. When the binding is modified, click **OK** to return to the **Alias Set** dialog box.
6. Repeat step 2 through step 5 for all bindings that you want to modify in this alias set.

Adding Bindings to an Alias Set

You can add one or more bindings to an alias set.

➤ To add a binding:

1. If have not already done so, in the **Metadata Navigator**, right-click the alias set to which you want to add bindings and select **Edit** to launch the **Alias Set** dialog box.
2. Complete step 2 through step 6 in "Creating Bindings Manually for an Alias Set " on page 148 for all bindings that you want to add to the selected alias set.

Deleting Bindings from an Alias Set

You can delete bindings from an alias set.

➤ To delete a binding:

1 **If have not already done so, in the Metadata Navigator, right-click the alias set from which you want to delete bindings and select Edit to launch the Alias Set dialog box.**

2 **Select the bindings 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 multiple consecutive bindings from the list. Use the Ctrl key to select multiple nonconsecutive bindings from the list.

3 **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

You can perform these tasks on selected alias sets:

- Copy—See "Copying or Moving Alias Sets" on page 152.
- Rename—See "Renaming Alias Sets" on page 153.
- Delete—See "Deleting Alias Sets" on page 153.
- Export—See "Exporting Alias Sets" on page 153

Copying or Moving Alias Sets

➤ To copy an alias set:

1 **In the Metadata Navigator, right-click the alias set to copy and select Copy.**

2 **Right-click the destination folder in the Metadata Navigator and select Paste from the popup menu.**

If you are copying to a different folder in the **Metadata Navigator**, the alias set name is retained.

If you are copying to the same folder, the alias set is automatically renamed for you. For example, an alias set named `skuAS` is automatically renamed `skuAS_copy` when copied to the same folder.

3 **View the copied alias set in the Metadata Navigator.**

➤ To move an alias set to a different folder:

1 **In the Metadata Navigator, select the alias set to move.**

2 **Drag the alias set and drop it on the destination folder.**

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 Metadata Navigator, right-click the alias set to rename and select Rename.**

2 **White the alias set name is highlighted in the Metadata Navigator, type the new name.**

3 **View the renamed alias set in the Metadata Navigator.**

Deleting Alias Sets

➤ To delete an alias set:

1 **In the Metadata Navigator, right-click the alias set to delete and select Delete.**

Note: Use the Shift key to select multiple consecutive alias sets to delete from the tree. Use the Ctrl key to select multiple nonconsecutive alias sets to delete from the tree.

2 **At the Confirm Delete prompt, click Yes.**

The alias sets you selected for deletion are cleared from the **Metadata Navigator**.

Note: If you do not want to be prompted to confirm deletions, select the "Do no show me this message again" check box.

Exporting Alias Sets

You can export selected alias sets to an XML file, as described in "Exporting Selected Catalog Elements" on page 60. The XML file can then used to import the alias sets into a catalog on another machine.

See "Exporting and Importing the Essbase Studio Catalog Database" on page 58 for all export and import topics.

9 Hierarchies

In This Chapter

About Hierarchies

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. Any dimension can be used to consolidate data for reporting purposes. Levels in the hierarchy 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.

The hierarchies used in a cube schema 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 CITY is 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 only include 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, in a cube that does not support duplicate member names, if CITY is a child of STATE in your hierarchy, you can ensure unique member names by prefixing each CITY value with an appropriate value from the STATE column. You can then differentiate between CA_ALBANY and NY_ALBANY.

After a hierarchy is created, it can be dragged from the **Metadata Navigator** directly into the Cube Schema Wizard. Further, hierarchies can be reused in any number of cube schemas. By creating different 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 hierarchies:

- "Creating Standard and Measure Hierarchies" on page 156
- "Creating Calendar Hierarchies" on page 166

See also "Hierarchies Usage Guidelines and Limitations" on page 357.

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 159.

Illustrations of different types of hierarchies are presented in "Hierarchy Examples" on page 161.

To create calendar hierarchies, see "Creating Calendar Hierarchies" on page 166.

➤ To create a standard or measure hierarchy:

1 **Right-click the appropriate folder in the Metadata Navigator and select New, then Hierarchy or New, then Measure Hierarchy to launch the hierarchy editor.**

2 **Enter a Name for the hierarchy.**

The default name is `NewHierarchy` for standard hierarchies, and `NewMeasureHierarchy` for measure hierarchies.

3 **Optional: 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 **Data Source Navigator**. 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 first metadata element that 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 that 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 **Data 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 necessary for the hierarchy that you want to build.

Drag-and-drop from the **Data Source Navigator**:

a. In the **Data Source Navigator**, navigate to the first physical element that 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 that 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 that 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 necessary for the hierarchy you want to build.

Use the **Add** button:

a. Click the **Add** button and perform an action:

   - Select **Add child** or **Add sibling**.

     The **Select Entity** dialog box is displayed. Navigate to the dimension 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**.

     Under **Hierarchy**, in the editable field in the grid, enter the name of the user-defined member. The default name is "`NewMember`*N*."

     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. Valid selections are **Add child**, **Add sibling**, or **Add user-defined sibling**.

c. **Optional:** Repeat step 4.b as necessary for the hierarchy that 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 176. 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 138).**
2. **Right-click the appropriate folder in the Metadata Navigator and select New, and then 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  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 161
- "Multichain Hierarchies" on page 161
- "Multichain Hierarchy with a Shared Member (Alternate Hierarchy)" on page 162
- "Multichain Hierarchy with Attribute Dimensions" on page 162
- "Recursive Hierarchies" on page 164
- "Time Hierarchies Built from DATE Type Metadata Elements" on page 165

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 physical and logical elements.

Figure 5 is an example of a simple single-chain hierarchy for Market. This hierarchy has 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

Hierarchy	Key Binding
REGION	connection : \'tbcSource'::'tbc.region'.'REGION'
STATE	connection : \'tbcSource'::'tbc.market'.'STATE'

Figure 6 is a single-chain, multigeneration hierarchy, meaning there are more than two levels in the hierarchy. Based on the FoodMart database, the levels in this hierarchy come from the same table. Note that the levels in a multigeneration hierarchy need not come from the same table.

Figure 6 Single-chain, multigeneration hierarchy

Hierarchy	Key Binding
warehouse_country	connection : \'foodmartSource'::'xolapfoodmart.axolap.warehouse'.'warehouse_country'
warehouse_state_province	connection : \'foodmartSource'::'xolapfoodmart.axolap.warehouse'.'warehouse_state_province'
warehouse_city	connection : \'foodmartSource'::'xolapfoodmart.axolap.warehouse'.'warehouse_city'
warehouse_name	connection : \'foodmartSource'::'xolapfoodmart.axolap.warehouse'.'warehouse_name'

Multichain Hierarchies

A multichain 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.

Figure 7 Multichain hierarchy

Hierarchy	Key Binding
⊟ REGION	connection : \'sxrSource'::'sxr_rolap.sxr.REGION'.'REGION'
STATE	connection : \'sxrSource'::'sxr_rolap.sxr.MARKET'.'STATE'
⊟ COUNTRY	connection : \'sxrSource'::'sxr_rolap.sxr.SUPPLIER'.'COUNTRY'
STATE	connection : \'sxrSource'::'sxr_rolap.sxr.MARKET'.'STATE'

Multichain Hierarchy with a Shared Member (Alternate Hierarchy)

You can design a multichain hierarchy that builds dimensions with shared members, also known as alternate hierarchies.

One way to accomplish this is to 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. In Figure 8, this is the SKU_Diet. 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 (SKU_Diet) as the child, which will be the shared member, as shown in Figure 8.

Figure 8 Multichain hierarchy with shared member (alternate hierarchy)

Hierarchy	Key Binding
⊟ FAMILY	connection : \'tbcSource'::'tbc.family'.'FAMILY'
SKU	connection : \'tbcSource'::'tbc.product'.'SKU'
⊟ Diet	"Diet"
SKU_Diet	connection : \'tbcSource'::'tbc.product'.'SKU'

Multichain Hierarchy with Attribute Dimensions

To produce attribute dimensions in your cube, build a multichain hierarchy, adding the same dimension element or physical column as the leaf-level member of each chain, as shown in Figure 9.

Figure 9   Multichain hierarchy to build attribute dimensions

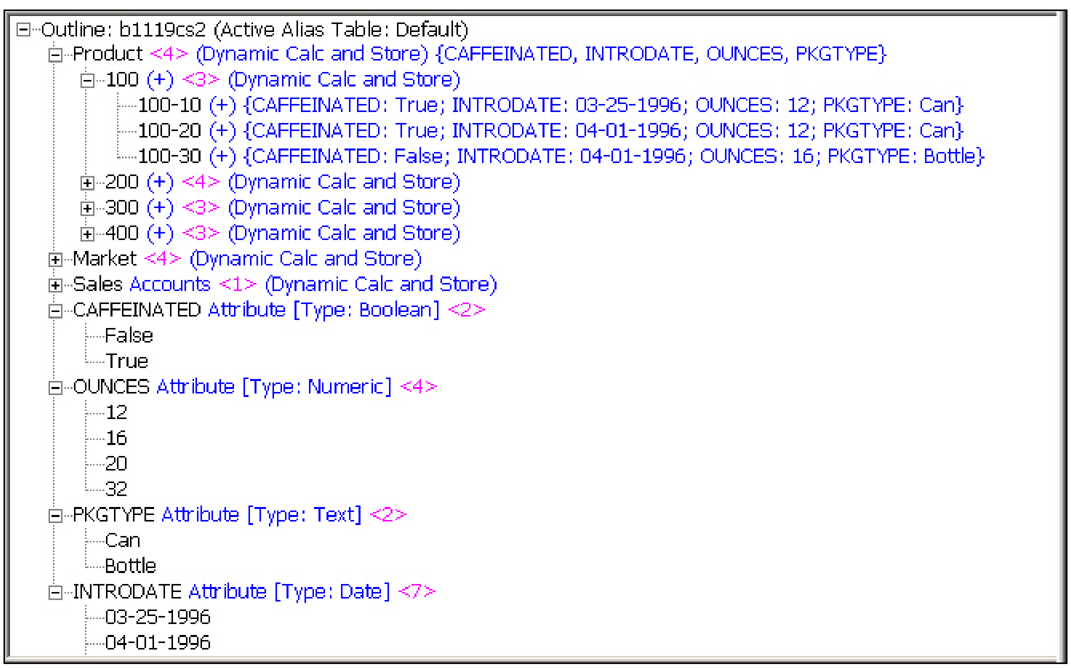
Hierarchy	Key Binding
⊟ FAMILY	connection : \'tbcSource'::'tbc.family'.'FAMILY'
SKU	connection : \'tbcSource'::'tbc.product'.'SKU'
⊟ CAFFEINATED	connection : \'tbcSource'::'tbc.product'.'CAFFEINATED'
SKU	connection : \'tbcSource'::'tbc.product'.'SKU'
⊟ OUNCES	connection : \'tbcSource'::'tbc.product'.'OUNCES'
SKU	connection : \'tbcSource'::'tbc.product'.'SKU'
⊟ PKGTYPE	connection : \'tbcSource'::'tbc.product'.'PKGTYPE'
SKU	connection : \'tbcSource'::'tbc.product'.'SKU'
⊟ INTRODATE	connection : \'tbcSource'::'tbc.product'.'INTRODATE'
SKU	connection : \'tbcSource'::'tbc.product'.'SKU'

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 the 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 220 for more information.

To set attribute formatting rules for the model, see “Defining Attributes in Models” on page 193.

Figure 10 shows a portion of the resulting Essbase outline after cube deployment.

Figure 10



Recursive Hierarchies

A recursive hierarchy contains members that are based on the contents of the two columns of a parent-child relationship.

Note: If a source database contains recursive tables, you must create a self-join between the parent and child columns to access related information and to consolidate the data properly. This can be done in the minischema in Essbase Studio. See "Adding or Editing Joins in a Minischema" on page 116 for more information.

In the TBC sample database, the Measures table contains two columns, "PARENT" and "CHILD", which have the parent-child relationship necessary to build a recursive hierarchy. The relationship between the rows of the PARENT and CHILD columns are illustrated below:

```
PARENT            CHILD
Measures          Profit
Profit            Margin
Margin            Sales
Margin            COGS
Profit            Total Expenses
Total Expenses    Marketing
Total Expenses    Payroll
Total Expenses    Misc
Measures          Inventory
Inventory         Opening Inventory
Inventory         Additions
Inventory         Ending Inventory
Measures          Ratios
Ratios            Margin %
Ratios            Profit %
Ratios            Profit per Ounce
```

Using the PARENT and CHILD columns above, an Essbase outline would be created as shown:

```
Measures
   Profit
      Margin
         Sales
         COGS
      Total Expenses
         Marketing
         Payroll
         MISC
   Inventory
      Opening Inventory
      Additions
      Ending Inventory
   Ratios
      Margin %
      Profit %
      Profit per Ounce
```

Remember, to build a recursive hierarchy properly, you must create a self join between the parent and child columns in the recursive table. Using the TBC sample, create a self join between the PARENT and CHILD columns of the Measures table.

Figure 11 is an example of the recursive hierarchy created in Essbase Studio using the PARENT and CHILD columns in the Measures table of the TBC sample database:

Figure 11 Recursive (parent-child) hierarchy

Hierarchy	Key Binding
⊟ PARENT	connection : \'tbcSource'::'tbc.measures'.'PARENT'
CHILD	connection : \'tbcSource'::'tbc.measures'.'CHILD'

Hierarchies Built from Physical and Metadata Elements

You can build hierarchies comprised of columns from the **Data Source Navigator** and columns from the **Metadata Navigator**. In Figure 12, the parent member, FAMILY, is a column that is dragged into the hierarchy from the **Data Source Navigator**; the child member, SKU, is a dimension element that is dragged into the hierarchy from the **Metadata Navigator**.

Figure 12 Mixed hierarchy created with physical and metadata elements

Hierarchy	Key Binding
⊟ FAMILY	connection : \'tbcSource'::'tbc.family'.'FAMILY'
SKU	connection : \'tbcSource'::'tbc.product'.'SKU'

When you drag a column from the **Data Source Navigator** into a hierarchy, a dimension element is automatically created for that column in the **Metadata Navigator**, in the same folder where the hierarchy is created.

Time Hierarchies Built from DATE Type Metadata Elements

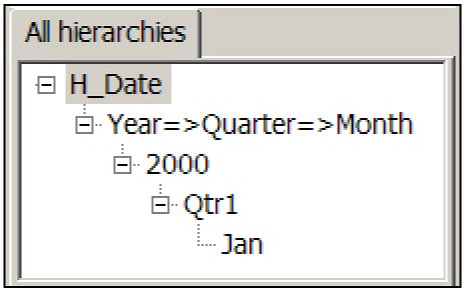
You must have metadata elements of the DATE type in order to build a time hierarchy like the one shown in Figure 13. Before you begin, ensure that you have completed the procedure in "Creating Date Elements" on page 142.

Figure 13 Time Hierarchy with Year, Quarter, and Month levels

Hierarchy	Key Binding
⊟ Year	'year'(connection : \'tbcSource'::'tbc.sales'.'TRANSDATE')
⊟ Quarter	'quarterAsString'(connection : \'tbcSource'::'tbc.sales'.'TRANSDATE')
⊟ Month	'monthShortName'(connection : \'tbcSource'::'tbc.sales'.'TRANSDATE')

Figure 14 shows the sample hierarchy you can view when clicking the Preview button in the hierarchy editor.

Figure 14   Preview of a time hierarchy



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 166.
- Fiscal—See "Fiscal Calendar Hierarchies" on page 167.
- Retail—See "Retail Calendar Hierarchies" on page 169.
- ISO—See "ISO Calendar Hierarchies" on page 170.
- Manufacturing—See "Manufacturing Calendar Hierarchies" on page 170.

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 179 for more information.

Gregorian Calendar Hierarchies

A Gregorian calendar is the standard 12-month calendar starting on January 1 and ending on December 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**, then **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 that you are modeling, and select a **First day of week.**

5 To define time depth, see "Defining Time Depth" on page 172.

Time depth applies to all calendar types.

6 To define day attributes, see "Defining Day Attributes" on page 172.

Day attributes apply to all calendar types.

7 To set linked value attributes, see "Linked Value Attributes" on page 173.

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 12-month reporting period includes two months of four weeks and one month of five 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. 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**, then **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**, or **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.
- **Ending week - number in month**—Select the **Month** and **Week** in which your fiscal year ends.
- **Year starts on a week that includes a specific date**—Select the specific date from **Month** and **Day**. The fiscal year begins on the first day of this week containing this date.
- **Year ends on a week that includes a specific date**—Select the date from **Month** and **Day**. The fiscal year ends on the last day of the week containing this date.
- **Year starts on the week on or immediately following a specific date**—Select the date from **Month** and **Day**. 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 **Month** and **Day**. 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 unavailable depending on the year semantic rule selections.

- **Starting week number specified**—The starting week number is specified by the selection in **Week** above.
- **Month always starts on a week that includes a specific date**—The date is specified by the selection in **Day** above.
- **Month starts on the week on or immediately following a specific date**—The date is specified by the selection in **Day** 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 that 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 **Day** above.

Click **OK** to return to the **Edit Calendar Hierarchy** dialog box.

5 In **Modeling Parameters**, enter the start and end dates of the period that you are modeling, and select a **First day of week.**

6 To define time depth, see "Defining Time Depth" on page 172.

Time depth applies to all calendar types.

7 To define day attributes, see "Defining Day Attributes" on page 172.

Day attributes apply to all calendar types.

8 To set linked value attributes, see "Linked Value Attributes" on page 173.

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 standard practice to omit the first week of a 53-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**, then **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 choose only **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 that includes the extra week.

Click **OK** to return to the **Edit Calendar Hierarchy** dialog box.

5 In **Modeling Parameters**, enter the start and end dates of the period that you are modeling, and select a **First day of week.**

6 To define time depth, see "Defining Time Depth" on page 172.

Time depth applies to all calendar types.

7 To define day attributes, see "Defining Day Attributes" on page 172.

Day attributes apply to all calendar types.

8 To set linked value attributes, see "Linked Value Attributes" on page 173.

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 (January 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 include only 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**, then **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 "Defining Time Depth" on page 172.

Time depth applies to all calendar types.

6 To define day attributes, see "Defining Day Attributes" on page 172.

Day attributes apply to all calendar types.

7 To set linked value attributes, see "Linked Value Attributes" on page 173.

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 13-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 53 weeks.

When you define the 13 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 or 53 weeks.

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- or 53-week year.

➤ To create a manufacturing calendar hierarchy:

1 Right-click the appropriate folder in the **Metadata Navigator** and select **New** and then **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 manufacturing 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 manufacturing year starts.
- **Ending week - number in month**—Select the **Month** and **Week** in which your manufacturing year ends.
- **Year starts on a week that includes a specific date**—Select the date from **Month** and **Day**. The manufacturing year begins on the first day of the week containing this date.
- **Year ends on a week that includes a specific date**—Select the date from **Month** and **Day**. The manufacturing year ends on the last day of the week containing this date.
- **Year starts on the week on or immediately following a specific date**—Select the date from **Month** and **Day**. The manufacturing 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 **Month** and **Day**. The manufacturing 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 manufacturing 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 manufacturing year.

Period Semantic Rules

Some of these options may be unavailable depending on the year semantic rule selections.

- **Quarter having 4 periods**—Select a quarter.
- **Period having 5 weeks**—Select a period.

Click **OK** to return to the **Edit Calendar Hierarchy** dialog 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 "Defining Time Depth" on page 172.**

Time depth applies to all calendar types.

7 **To define day attributes, see "Defining Day Attributes" on page 172.**

Day attributes apply to all calendar types.

8 **To set linked value attributes, see "Linked Value Attributes" on page 173.**

Linked value attributes apply to all calendar types.

9 **Click OK to close the Edit Calendar Hierarchy dialog box.**

Defining 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.**

Note the following:

- Unavailable levels with the check box selected are required and must be used with this calendar type.
- Unavailable 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 Edit Labeling Rules, select a rule from the list and click OK.**

Defining 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:

```
Week,Sunday
```

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 yo 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 156, starting at step 2.
    - If you are editing a calendar hierarchy, see "Creating Calendar Hierarchies" on page 166, select the appropriate calendar type, and follow the procedure for that calendar type, starting at step 2.

10 Cube Schemas

In This Chapter

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 a 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.

➤ Perform an action:

- To create a new cube schema, in the **Metadata Navigator**, select **New**, then **Cube Schema** to launch the **Cube Schema Wizard**.
- To edit a cube schema, right-click its name in the **Metadata Navigator** and select **Edit**.

The Cube Schema Wizard comprises two default screens, where you accomplish the following tasks:

- Choose measures and hierarchies—See "Choosing Measures and Hierarchies for a Cube Schema" on page 176.
- Set cube schema options—See "Setting Cube Schema Options" on page 178.

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 179 for more information.

Choosing Measures and Hierarchies for a Cube Schema

Choose measures and hierarchies whether you are creating or 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 a **Comment.**
3. Drag hierarchies that you want to include in this cube schema from the **Available Dimension Elements** list and drop them in the **Hierarchies** box 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 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** box on the right.

   You cannot add the same measure or measure hierarchy more than once.

   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: You may add only one measure hierarchy to the cube schema, but you may add one or more measures. When you add one or more individual measures that are not organized into a hierarchy, these are referred to as “loose” measures. You must add either a single measure hierarchy or one or more loose measures.

   Note: To remove 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 measure hierarchies** button.
5. **Optional:** Click **Preview Hierarchies** to view the structure of the hierarchies that you have chosen for this cube schema, as described in “Previewing Hierarchies” on page 176.

   When you are finished viewing hierarchies, click **OK**.
6. Click **Next** to open the next page of the wizard, **Cube Schema options**, as described in “Setting Cube Schema Options” on page 178.

Previewing Hierarchies

You can preview the structure of a hierarchy while creating or editing a hierarchy and while creating or editing a cube schema.

You can also preview a hierarchy anytime from the **Metadata Navigator**.

➤ To preview hierarchies:

1 **Access the Sample Data window by performing an action:**

- In the hierarchy editor, when creating or editing a hierarchy, click **Preview**.
- 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 preview 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**.

2 **Choose a viewing option:**

- **With Caption Binding**—displays the hierarchies using data derived from the caption binding expression of each dimension element in the hierarchy.
- **With Key Binding**—displays the hierarchies using data derived from the key binding expression of each dimension element in the hierarchy.

The **Sample Data** window is displayed.

For example, using the TBC sample database, in a simple Market hierarchy made up of these members:

- REGION, where the caption binding expression equals `connection : \'tbcSource'::'tbc.region'.'REGION' . toString`; and key binding expression equals `connection : \'tbcSource'::'tbc.region'.'REGIONID' . toString`
- STATE, where the caption binding expression equals `connection : \'tbcSource'::'tbc.market'.'STATE' . toString`; and the key binding expression equals `connection : \'tbcSource'::'tbc.market'.'STATEID' . toString`

When you select the **With Caption Binding** option, the preview results are:

```
Market
  REGION=>STATE
    East
      New York
      Massachusetts
      Florida
      Connecticut
      New Hampshire
    West
      California
      Oregon
      etc.
```

When you select the **With Key Binding** option, the preview results are:

```
Market
  REGION=>STATE
    1
      1
      2
      3
      3
      5
    2
      6
      7
      etc.
```

Note: When you select a Performance Management Architect hierarchy to preview, the **Preview EPMA Dimension** window is displayed where you will view the properties of individual members (see step 4).

3 If previewing hierarchies from the **Choose Measures and Hierarchies** page of the **Cube Schema Wizard**, perform these steps:

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 hierarchy sample, click **OK** to return to the **Cube Schema Wizard**.

4 If previewing a Performance Management Architect dimension, in the **Preview EPMA 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 4.b for each member whose properties you want to view.

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** page, 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 179 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 complete these steps:

a. Enter a **Model Name**.

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 **For Performance Management Architect-based models only:** In **Model Type**, select whether data storage type for this model will be **Aggregate Storage** or **Block Storage**.

Note: You must choose the data storage type during the cube schema creation process. You cannot change it in the Essbase model.

4 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 179.

Note: If you encounter an out-of-memory error when working with a cube schema, increase the virtual memory setting for Essbase Studio Console, as described in "Configuring Virtual Memory" on page 62.

5 **Optional:** When the process is complete, review the graphical representation of the Essbase model in the Essbase Studio Console work area.

Defining Data Load Mappings

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. Use this dialog box to override the default load bindings that will be used for loading data.

These guidelines apply to data load mapping:

- Data is loaded at the leaf level of the hierarchy, including recursive hierarchies; therefore, you define the data load mapping at the lowest level of the hierarchy.

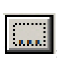
- For a pure user-defined multichain hierarchy, only one default load binding can be specified at the leaf level for the entire hierarchy. Check boxes are enabled in the Data Load Binding column, and you can select only one check box per pure user-defined hierarchy.
- For mixed multichain hierarchies, that is, hierarchies that contain relational chains and user-defined chains:
  - One data load binding can be specified for each unique leaf level.
  - The check box for the leaf level of all user-defined hierarchy chains can be selected.
- For a measure hierarchy, there is no restriction on the number of bindings that can be specified. They only condition is that they are specified at the leaf level.
- For a calendar hierarchy, you must complete the procedure in this topic. 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.
- 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.**
2. **Note the entries in the Key Binding column.**

   The **Key Binding** column contains:

   - Key bindings for elements whose data load binding can be changed.
   - The definition of user-defined members.
   - The status “Delayed” for any dimension elements that have a **Delayed** key binding.

   The key bindings, definitions, and status in this column are read-only and are presented for your reference.
3. **In the Data Load Binding column, click in the row of the lowest level of a selected hierarchy and take the appropriate action :**
   - For relational, recursive, and calendar hierarchies whose key binding you want to change, click the ellipsis button, , to launch the **Default Load Binding** dialog box, and proceed to step 4.
   - For pure user-defined multichain hierarchies, select one check box.
   - For mixed multichain hierarchies, build a data load binding expression (step 4), or select the check box at the leaf level of one or more user-defined hierarchy chains.
   - For single or multichain measure hierarchies, specify a data load binding for one or more chains (step 4).

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 that 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 that 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 places.

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, multichain 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, multichain 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 multichain 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 cube schema creation, you can create one anytime from the existing cube schema.

➤ To create an Essbase model from a 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 a **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**.

11 Essbase Properties

In This Chapter

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 185
- **Alias** tab—"Selecting Alias Sets for an Essbase Model" on page 192
- **Attributes** tab—"Defining Attributes in Models" on page 193

For information on models, see "Essbase Models Overview" on page 184.

To access the **Essbase Model Properties** dialog box, see "Accessing the Essbase Model Properties Dialog Box" on page 185.

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 more information on Essbase models, see "Essbase Models Overview" on page 184.

You can access the **Essbase Model Properties** dialog from the **Metadata Navigator**. See "Accessing the Essbase Model Properties Dialog Box" on page 185.

Note: If you encounter an out-of-memory error while working with an Essbase model, increase the virtual memory setting for Essbase Studio Console, as described in "Configuring Virtual Memory" on page 62.

Essbase 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 an 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 199.

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 188 and "XOLAP Overview" on page 189.

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 199 and "Member Properties" on page 218.

The **Essbase Model Properties** dialog box has three tabs for editing model properties:

- **General** tab—"Setting General Model Properties" on page 185
- **Alias** tab—"Selecting Alias Sets for an Essbase Model" on page 192
- **Attributes** tab—"Defining Attributes in Models" on page 193

See "Accessing the Essbase Model Properties Dialog Box" on page 185.

Accessing the Essbase 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 "Essbase Models Overview" on page 184.

➤ To set general properties for an Essbase model:

1. Access the **Essbase Model Properties** dialog box using the procedure in "Accessing the Essbase Model Properties Dialog Box" on page 185.

   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:** Click the button to right of the **Last deployed** field to view the complete deployment history of the model.

   The **Last deployed** field displays information about the most recent deployment.

   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 189.

5 **Optional:** Click the **Custom data load settings** button to access the **Define Data Load 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 187 for more information.

6 Under **Typed Measures Support**, select a **Date format.**

The default date format is `yyyy-mm-dd`.

See "Specifying Typed Measures" on page 235 for more information.

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**.

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 187 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 Essbase Model Properties Dialog Box" on page 185.
2. Click the **Custom data load settings** button to access the **Define Data Load Settings** dialog box.
3. Select **Use Custom SQL for data load.**
4. **Optional:** To see the SQL statements as a folder, select the **View statements as folder** option.

   By default, the SQL statements in the left text box are displayed as a list.
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**.

   Note: If you have edited the data load SQL, and then want to use the standard data load SQL, simply clear the "Use Custom SQL for data load" check box. Your customized SQL is retained, but not used.

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.
- If you have selected the "Use Custom SQL for data load" check box, and you have not copied over or edited one or more SQL statements, then data load is ignored for those statements. This allows you to exclude statements from data load.

  To keep the SQL for a particular statement in a custom data load without editing it, copy it over to the editing pane while the **Use Custom SQL for data load**check box is not selected.
- 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.
- 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.
- If a data source column is bound to a level-0 member set, and contains both base members and shared members, it is possible that the data aggregated in the Essbase Studio-generated data load SQL for the specific base members (which have shared members in the same dimension) will be augmented at times, depending on the number of shared members. In this case, Oracle recommends that users define their own custom data load SQL as described in "Overriding Standard Data Load SQL" on page 187. This behavior is true both for Essbase models created from metaoutlines imported from Oracle Essbase Integration Services, and for Essbase models created from the beginning in Essbase Studio.

  For example, using the TBC sample, the data aggregated for 100-10 in the standard data load SQL is twice as much as it should be (doubled) because 100-10 has shared members under "Diet". To get the correct aggregated data, users can select the **Use Custom SQL for data load** option, and manually add one more "group by" column, "Family", to the custom data load SQL.

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 188.

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 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 191.
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

See also "XOLAP Functionality Guidelines" on page 360.

XOLAP has several restrictions and several unsupported usages.

Restrictions for XOLAP

XOLAP has the following restrictions:

- Although the **Outline Editor** in Administration Services Console does not prevent you from modifying an XOLAP outline, it is not allowed. XOLAP operations do not automatically incorporate changes in the structures and the contents of the dimension tables after an outline is created. To modify an outline, you must modify the underlying Essbase model and redeploy to a new Essbase database.
- Incremental builds for XOLAP-enabled models are not supported.

  If you are deploying from an Essbase model that is enabled for XOLAP, it is strongly recommended that you use a new application and database name; or use the **Delete all members first** option to deploy over an existing XOLAP application.
- When derived text measures are used in cube schemas to build an Essbase model, XOLAP cannot be enabled for the model.
- XOLAP can be used only with aggregate storage. The database is automatically duplicate-member enabled.

- Essbase Studio can create attribute dimensions in Essbase models enabled for XOLAP; however, attribute dimensions must have only one child level.
- 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.
- User-defined members in an XOLAP model must not have stored parent or children members. The hierarchy must be specified as a dynamic hierarchy.
- To compute their values in XOLAP, user-defined members must have valid formulas associated with them.
- On the **General** tab of **Essbase Model Properties** dialog box, the **XOLAP Model** option is disabled for models created from dimension server (Performance Management Architect) sources.

Usages Not Supported in XOLAP

XOLAP does not support the following usages:

- Flat files
- Ragged hierarchies
- Recursive hierarchies
- Calendar hierarchies
- Filters
- Text measures
- Multiple relational data sources
- Dimension server (Performance Management Architect) data sources

Designating a Model for XOLAP

You can designate a model to be enabled for XOLAP.

For information on models, see "Essbase Models Overview" on page 184.

For information on XOLAP, see "XOLAP Overview" on page 189.

➤ To designate a model for XOLAP:

1 Access the **Essbase Model Properties** dialog box using the procedure in "Accessing the Essbase Model Properties Dialog Box" on page 185.

  The model and cube schema names are displayed in the read-only **Name** and **Cube Schema** fields.

2 Select the check box **XOLAP Model.**

3 Click **Apply.**

Selecting Alias Sets for an Essbase Model

Select alias sets to use in the model in the **Alias** tab of the **Essbase Model Properties** dialog box.

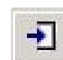
For information on aliases, see “Aliases Overview” on page 237.

For information on models, see “Essbase Models Overview” on page 184.

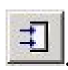
Note: If you plan to optimize data loads (see “Optimizing Data Loads” on page 217), one alias table is required by Essbase to hold the aliases that will be used in data load alias optimization. You can select 31 other tables as alias sets for a total of 32. If you select more than 31 tables, some of the tables will be dropped during data loads.

➤ To select alias sets to use in an Essbase model:

1. Access the **Essbase Model Properties** dialog box using the procedure in “Accessing the Essbase Model Properties Dialog Box” on page 185.
2. With focus on the Essbase model name, select the **Alias** tab.
3. To move tables from the **Available Tables** list to the **Selected Tables** list, perform an action:

   - Select one or more alias sets, and click **Add the selected alias sets to the model**, .

     Note: You can also double-click on a table to move it.

   - Click **Add all available alias sets to the model**, .

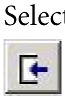
4. If you have selected alias sets with duplicate names in the **Name in Cube** column, a message prompts you to rename those alias sets in the **Metadata Navigator**.

   Because alias sets can have the same name under different folders, providing a new (and different) name for the alias set makes all alias table names unique inside a cube.

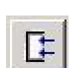
5. **Optional:** Rearrange the order of the alias sets in the **Selected Tables** box by selecting a table and then clicking one of the following buttons:

   - **Move the selected alias up in the list**, 
   - **Move the selected alias down in the list**, 

6. To move tables from the **Selected Tables** list to the **Available Tables** list, perform an action:

   - Select one or more alias sets, and click **Remove the selected alias sets from the model** 

     Note: You can also double-click on a table to move it.

   - Click **Remove all alias sets from the model**, 

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**.

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 193.

For information on varying attributes, see "Varying Attributes Overview" on page 194.

The following topics discuss the Essbase model properties that you edit in the **Attributes** tab:

- "Setting Attribute Member Names Format" on page 195
- "Setting Attribute Calculations Member Names Format" on page 196
- "Specifying Attribute Boolean, Date, and Numeric Ranges" on page 197

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 calculation 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 *Oracle 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
    |
    |_Color
       |
       |__SUBSKU
```

When Color is set as a varying attribute in Essbase Studio, the retrieval results would be similar to the following table:

Table 4 Varying Attribute Example of Color Associated with SUBSKU

SUBSKU	SKU
Red	100
White	400
White	600
Black	200
Black	300
Silver	500

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 Essbase Model Properties Dialog Box” on page 185.

2 Select the **Attributes** tab.

3 Locate the **Member name format** group.

4 Select a **Value** to be used as a prefix or suffix:

Note: You can attach both a prefix and a suffix to a member name.

- **None**
- **Parent**
- **Grandparent**
- **Ancestor**
- **Dimension**

5 Select a **Separator** to be used with the prefix or suffix:

- **Underscore**
- **Pipe**
- **Caret**

6 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.

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**.

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 197

➤ 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 Essbase Model Properties Dialog Box” on page 185.

2 Select the **Attributes** tab.

3 Locate the **Attribute calculations member names** group.

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 Essbase Model Properties Dialog Box" on page 185.

2 Select the **Attributes** tab.

3 Locate the **Boolean, date, and numeric ranges** group.

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:

- **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.
- **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 201
- **Info** tab—"Editing Dimensions" on page 203
- **Account Info** tab—"Editing Account Dimensions Properties" on page 208
- **Formula** tab—"Adding Formulas to Dimensions" on page 211
- **Alias** tab—"Displaying and Editing Dimension Aliases" on page 212

- **UDAs** tab—"Assigning User-Defined Attributes to Dimensions" on page 213
- **Outline Build** tab—"Selecting Outline Build Options" on page 214

For information on dimensions, see "Dimensions Overview" on page 199.

To access the dimension properties tabs, see "Accessing the Dimension Properties Tabs" on page 201.

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 199
- "Rules for Using Time Dimensions" on page 200
- "Rules for Using Accounts Dimensions" on page 200

To access the dimension properties tabs, see "Accessing the Dimension Properties Tabs" on page 201.

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 200.

- 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 200.

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 206.

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 206.

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 naming generations and levels.

For information on dimensions, see "Dimensions Overview" on page 199.

➤ 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 201****.**

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 368.

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 **Optional: To create or edit generation or level names, click the Edit button.**

A status message next to the **Edit** button tells you whether generations or levels have been named.

Proceed to "Naming Generations and Levels" on page 202 for more information.

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**.

Naming Generations and Levels

You can create your own names for generations and levels in an Essbase model. The name is a word or phrase that describes the generation or level. For example, you might create a generation name called "Cities" for all cities in the outline. You can define only one name for each generation or level.

Use generation and level names in calculation scripts and report scripts wherever you need to specify either a list of member names or a list of generation or level numbers. For example, you can limit a calculation in a calculation script to the members of a specific generation.

In a dimension that allows duplicate member names, you can specify that unique member names are required for a particular generation or level.

➤ To create, modify, or delete a generation name or level name:

1 **Select the appropriate dimension, then access the dimension properties tabs of the Essbase Model Properties dialog box using the procedure in "Accessing the Dimension Properties Tabs" on page 201.**

2 **In the dimension properties General tab, under Named Generations/Levels, click the Edit button.**

The **Edit Named Generations and Levels** dialog box is displayed.

3 **To create a generation or level name, complete these tasks:**

a. Click the **Add a named generation** button or **Add a named level button** button.

A default generation or level name and number is provided.

b. To change the default name, click in the **Generation Name** or **Level Name** cell and type the new name.

c. In the **Number** column of the same row, click in the cell and type a generation number or level number.

For example, to name a generation for the months in the Sample Basic database, select 3. To name a level for the months in the Sample Basic database, select 1.

d. **Optional:** In Essbase models that allow duplicate member names, to require unique member names within a particular generation or level in a duplicate member name dimension, select the check box in the **Unique** column.

e. Repeat step 3.a through step 3.e for all named generations or levels you want to create.

4 **Optional: Click Organize named generations by value or Organize named levels by value to view the items you have added in order by generation or level number.**

5 **To modify a named generation or level, complete step 3.b through step 3.e.**

6 **To delete a named generation or level, complete an action:**

- Select one or more named generations to delete, then click the **Delete the selected named generation(s)**

- Select one or more named levels to delete, then click the **Delete the selected named level(s)**

7 Click **OK** to exit the dialog box and return to the dimension properties **General** tab.

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 199

The following topics describe the dimension properties you edit in the **Info** tab:

- "Selecting the Dimension Type" on page 203
- "Using Dynamic Time Series" on page 204
- "Selecting the Dimension Storage Method" on page 205
- "Selecting Two Pass Calculation" on page 205
- "Selecting the Data Storage Method" on page 206
- "Selecting the Solve Order" on page 207

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 199.

➤ To select a dimension type:

1 Access the dimension properties tabs of the **Essbase Model Properties** dialog box using the procedure in "Accessing the Dimension Properties Tabs" on page 201.

2 Select the **Info** tab.

3 Locate the **Dimension Type** group.

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 200.

  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 200.

  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 204.

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 201.

2 Select the **Info** tab.

3 In the **Dimension Type** group, select **Time.**

4 Click the **Dynamic series** button.

5 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

6 Click in the **Generation** column, and select a generation level from the drop-down list.

7 Enter one or more aliases for the time series member, if you require.

Note: The column of alias sets is displayed only when alias sets have been added to the model (see "Creating Alias Sets" on page 148).

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.

8 Click **OK**.

9 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 201.

2 Select the **Info** tab.

3 Locate the **Dimension Storage** group.

4 Select a **Dimension Storage** option:

- **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 206.

➤ To select Two Pass Calculation:

1 Access the dimension properties tabs of the **Essbase Model Properties** dialog box using the procedure in "Accessing the Dimension Properties Tabs" on page 201.

2 Select the **Info** tab.

3 Locate the **Two Pass Calculation** group.

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:

1 Access the dimension properties tabs of the **Essbase Model Properties** dialog box using the procedure in "Accessing the Dimension Properties Tabs" on page 201.

2 Select the **Info** tab.

3 Locate the **Data Storage** group.

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 229.

- **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 227.

➤ To select a solve order:

1 **Access the dimension properties tabs of the Essbase Model Properties dialog box using the procedure in "Accessing the Dimension Properties Tabs" on page 201.**

2 **Select the Info tab.**

3 **Locate the Solve Order grouping.**

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 199.

The following topics describe the dimension properties that you edit in the Account Info tab:

- "Selecting a Time Balance" on page 208
- "Selecting the Variance Reporting Method" on page 209
- "Selecting the Skip Option" on page 210

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 209.

➤ To select a time balance:

1 Access the dimension properties tabs of the **Essbase Model Properties** dialog box using the procedure in "Accessing the Dimension Properties Tabs" on page 201.

2 Select the accounts dimension.

3 **Select the Account Info tab.**

4 **Locate the Time Balance grouping.**

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 210.

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 Qtr1 to reflect the beginning inventory value for Qtr1, set a time balance property of first on the accounts member Opening Inventory. When Opening Inventory is calculated across the Year dimension, Qtr1 becomes equal to Jan (the first child in the Qtr1 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:

1 Access the dimension properties tabs of the **Essbase Model Properties** dialog box using the procedure in "Accessing the Dimension Properties Tabs" on page 201.

2 Select the accounts dimension of the model.

3 Select the **Account Info** tab.

4 Locate the **Variance Reporting** group.

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:

1 Access the dimension properties tabs of the **Essbase Model Properties** dialog box using the procedure in "Accessing the Dimension Properties Tabs" on page 201.

2 Select the accounts dimension.

3 Select the **Account Info** tab.

4 Locate the **Skip** group.

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 "About Formulas" on page 211.

For information on dimensions, see "Dimensions Overview" on page 199.

➤ 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 201.**

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.

Note: 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**.

About Formulas

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, members, user-defined members, and measures.

Following are some guidelines for working with formulas:

- Formulas in block storage models must be in the standard Essbase calculation script format.

- In formulas, member names surrounded by quotation marks, such as `"Root Beer"` or `"Cream Soda"`, are handled differently depending on their source.
  - When entering a formula directly into the **Formula** text box in **Essbase Model Properties**, use this format:

    ```
    ""Root Beer"+"Cream Soda";"
    ```
  - When referencing an external source column in the **From External Source** field in **Essbase Model Properties**, ensure that in the relational source, a backslash (`\`) is entered before each quotation mark; for example:

    ```
    \"Root Beer\"+\"Cream Soda\"
    ```
  - When working with BSO cubes built in Performance Management Architect, in the **Member Formula BSO** field in Performance Management Architect surround the entire formula in quotation marks; for example:

    ```
    ""Root Beer"+"Cream Soda""
    ```

  If you do not follow the format guidelines above, the result is that Essbase strips out the opening and closing quotation marks, for example, `Root Beer"+"Cream Soda`, making the formula invalid.
- When working with aggregate storage models:
  - 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.
  - You can specify formulas only for members within a dynamic hierarchy.

For more information on working with formulas, see the *Oracle Essbase Database Administrator's Guide*.

Displaying and Editing Dimension Aliases

In the **Alias** tab, you can see a list of current aliases for a dimension.

See the following topics for other information about aliases and alias sets:

- "Aliases Overview" on page 237.
- "Creating Alias Sets" on page 148.
- "Dimensions Overview" on page 199.

➤ To display and edit the assigned alias sets for a dimension:

1. Access dimension properties tabs of the **Essbase Model Properties** dialog box using the procedure in "Accessing the Dimension Properties Tabs" on page 201.
2. Select the **Alias** tab.

   The list of aliases assigned to the dimension, if any, is displayed. The name shown in the **Alias Set** column maps to the **Name in Cube** specified in **Alias** tab for the Essbase model.

3 **Optional: Edit the entry in the Source Mapping column to update the dimension alias for a selected dimension.**

For example, for a dimension named "Product" using a French alias set of , you might enter "Produit" in the **Source Mapping** column.

4 **Repeat step 3 for each dimension whose alias source mapping you want to change.**

Note: To modify alias assignments, see "Selecting Alias Sets for an Essbase Model" on page 192.

You can edit dimension aliases by adding prefixes or suffixes to them. The procedures for doing this are the same as the procedures for adding prefixes or suffixes to member aliases.

➤ To add prefixes to dimension aliases, follow the procedure in "Adding Prefixes to Aliases" on page 242.

➤ To add suffixes to dimension aliases, follow the procedure in "Adding Suffixes to Aliases" on page 243.

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 214.

See "Dimensions Overview" on page 199.

➤ To assign a UDA 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 201.**

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, consider the following guidelines:

- 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 "About Hierarchies" on page 155.

For information on dimensions, see "Dimensions Overview" on page 199.

The following topics describe the dimension properties you edit in the **Outline Build** tab:

- "Selecting Hierarchy Storage Settings" on page 214
- "Moving Duplicate Member Settings" on page 215
- "Placing Actual Members Before Shared Members" on page 216
- "Optimizing Data Loads" on page 217

Selecting Hierarchy Storage Settings

You can specify storage properties for hierarchies and alternate hierarchies in the building of Essbase outlines.

➤ To specify the storage option for hierarchies:

1. Access the dimension properties tabs of the **Essbase Model Properties** dialog box using the procedure in "Accessing the Dimension Properties Tabs" on page 201.
2. Select the **Outline Build** tab.
3. In the **Hierarchy settings** group, select an option:

- **Stored at dimension level**—Sets the dimension as a stored hierarchy. This is the default.
- **Dynamic at dimension level**—Sets the dimension as a dynamic hierarchy. When selected, the **Create as compression dimension** check box is enabled.

  Optional:

  Create as compression dimension check box—Optimizes an aggregate storage database.

  When selected, data is grouped in the dimension as `(key, multiple values)`.

  If this check box is not selected, data is stored as `(key, single value), (key, single value)`.

  Tip: Oracle recommends optimizing aggregate storage databases by compressing the accounts dimension.

  For further information on the compression dimension, see "Understanding the Compression Dimension for Aggregate Storage Databases" in the *Oracle Essbase Database Administrator's Guide*.
- **Multiple-hierarchy enabled**—Uses both stored and dynamic hierarchies in the dimension.

  Note: Because hierarchies determine how data is consolidated and navigated, you should review any aggregate storage settings you may have for generation 2 members by clicking the link to the Member Info tab.

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**.

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 201.

2 Select the **Outline Build** tab.

3 In the **Move duplicate member settings** group, select an option:

- **Do not move duplicate members**—Duplicate members are not moved in the Essbase outline, but shared members are added.
- **Move duplicate members (and their descendants)**—Duplicate members and their descendants are moved in the Essbase outline.

- **Ignore duplicate members, do not move**—Duplicate members are not moved in the Essbase outline, and shared members are not added.
- **Move Gen2 duplicates only**—Only Generation 2 members and their descendants are moved in the Essbase outline
- **Move non-Gen2 duplicates only**—Generation 2 members are ignored and not moved in the Essbase outline.

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**.

Placing Actual Members Before Shared Members

For aggregate storage outlines, Essbase requires that actual members are always placed before shared members. When building your Essbase model, you may have placed shared members before actual members in recursive dimensions.

By selecting a check box in the **Outline Build** tab of Essbase Model Properties, all instances of shared members before actual members are reversed.

➤ If the Essbase model contains shared members placed before actual members, select this check box:

Reverse position of shared and actual members if shared member is located before actual member

Below are examples showing the recursive hierarchy members before and after Essbase fixes the shared member and actual member order. Note that the order of the parents remains the same; the position of the actual member changes.

Case 1, Before

```
Employees (dimension)
    Engineer
        John Smith (shared)
    Manager
        John Smith (shared)
    Director
        John Smith (actual)
```

Case 1, After

```
Employees (dimension)
    Engineer
        John Smith (actual)
    Manager
        John Smith (shared)
    Director
        John Smith (shared)
```

Case 2, Before

```
Employees (dimension)
    Engineer
        John Smith (shared)
        Paul Williams (shared)
    Manager
        John Smith (actual)
        Paul Williams (actual)
```

Case 2, After

```
Employees (dimension)
    Engineer
        John Smith (actual)
        Paul Williams (actual)
    Manager
        John Smith (shared)
        Paul Williams (shared)
```

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 Alias Sets for an Essbase Model" on page 192.

➤ 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 201.
2. Select the **Outline Build** tab.
3. Locate the **Optimize** data load group.
4. Select **Optimize data load.**

   Note: This option will be ignored if the data load bindings were overridden in the Cube Schema wizard.

Ordering Dimensions

Dimension sort order determines the order that dimensions appear in the Essbase model and subsequent Essbase outline and, in turn, impacts the order the that dimensions are calculated. Calculation order for aggregate storage and block storage databases should be determined in advance, when designing the database. Note that dimension order can affect performance, especially for block storage databases. Calculation of aggregate storage and block storage databases is described in the *Oracle Essbase Database Administrator's Guide*. However, keep in mind the following:

- Aggregate storage databases—The dimension solve order is used by aggregate storage databases and should be taken into account when setting the dimension sort order. If no dimension solve order or sort order is set, aggregations are performed in outline order. Attributes are calculated last.
- Block storage databases—Dimension sort order is based on the calculation order, which is intrinsic in Essbase. Dense dimensions (for example, the time or measures dimensions) are calculated first, sparse dimensions are calculated next, and attribute dimensions are calculated last.

Dimension sort order uses buttons on the Essbase model toolbar.

Figure 15 Dimension sort order toolbar buttons

➤ To sort dimensions in the Essbase model:

1 **Select a dimension to move, then click the up or down arrows to move the dimension's position within the model.**

- To move a dimension up in the model, click **Move the selected dimension up in the list** .

  Alternatively, use the keyboard shortcut Alt+Shift+Up arrow key.
- To move a dimension down in the model, click **Move the selected dimension down in the list**.

  Alternatively, use the keyboard shortcut Alt+Shift+Down arrow key.

2 **To save the changes in dimension order, click Apply the changes in dimension ordering to the model.**

3 **Optional: To cancel dimension order changes made during the current session, click Cancel the changes in dimension ordering.**

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 219
- **Info** tab—“Editing Member Information” on page 225
- **Account Info** tab—“Editing Members in Accounts Dimensions” on page 232
- **Formula** tab—“Adding Formulas to Members” on page 236
- **Alias** tab—“Editing Member Aliases” on page 237
- **UDAs** tab—“Assigning User-Defined Attributes to Members” on page 244

- **Transformations** tab—"Transforming Members" on page 244

For information on members, see "Members Overview" on page 219.

To access the member properties tabs, see "Accessing the Member Properties Tabs" on page 219.

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 219.

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 219.

The following topics describe the member properties you edit in the **General** tab:

- "Specifying General Member Properties" on page 220

- “Selecting Members as Attributes” on page 220
- “Setting Varying Attributes for Members” on page 222

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 219.

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 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 193.

➤ 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 219.

2 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.

3 Select the attribute type from the drop-down list:

- **Numeric**
- **Boolean**
- **String**

- **Date/Time**

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**.

Setting Up a History Table for Varying Attributes

You can set varying attributes for members. When you work with varying attributes, you should first prepare a table called the "history" table. Once you have set up the history table, you are ready to complete the procedure in "Setting Varying Attributes for Members" on page 222.

The history table may contain a column (or more for outlines that support duplicate members) for varying attribute members, and several columns whose values can be used by Essbase to locate members in independent dimension.

For example:

```
Product Table

|  ProductID  |  SKU  |  Other product columns  |

Fact Table (can also be used to build the Time Dimension)

|  ProductID  |  Transdate  |  Other fact table columns  |

    Product Package Table (history table for varying attributes [VA])

     |  ProductID  |  Pkg Type (VA)  |  From Date  |  To Date  |
     |  1          |  Plastic Bag    |  Jan        |  Jul      |
     |  1          |  Paper Bag      |  Aug        |  Dec      |
```

The above example involves a fact table that contains the column "Transdate," which can be used to build a Time dimension in Essbase Studio.

However, most Essbase Studio users do not build a Time dimension from the fact table. In fact, Oracle recommends that users NOT build any dimensions directly from the fact table. Therefore, the table schema would look like this:

```
Employee Table

|  EmpID  |  EmpName  |  SS#  |

Time Table (to be used to build the Time dimension

|  Columns for Date Time...  |  TimeID  |

Fact Table (can also be used to build Time dimension)

|  EmpID  |  Hire Date (TimeID)  |  Other fact table columns  |

    Position Table (history table for varying attributes)

     |  EmpID  |  Position (VA)  |  From Date  |  To Date  |
```

The “Position” table has joins to the Employee and Time tables.

The dimension build query for the “Position” dimension involves the Time table. Since it is much smaller than fact table, it should not have a major impact on build performance.

If the “To” column is empty, the value of the “From Date” column can be the TimeID column.

```
Position Table (the history table for varying attributes)

 |  EmpID  |  Position (VA)  |  From Date (TimeID)  |
```

Setting Varying Attributes for Members

You can set varying attributes for members. Before you set varying attributes, Oracle recommends you set up a history table, as described in “Setting Up a History Table for Varying Attributes” on page 221.

For information on varying attributes and their use, see “Varying Attributes Overview” on page 194.

Note: 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 219.
2. 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.
3. From the drop-down list, select the **Attribute type**:
   - Numeric
   - Boolean
   - String
   - Date/Time

   Note: The default attribute type is String.
4. In the Varying Attribute Settings grouping, click the **Edit** button.

   The **Edit Varying Attributes** dialog box is displayed.
5. Select the **Create as Varying Attribute** check box.
6. 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.

7 **Locate the Independent Dimension Settings group.**

8 **In the Dimension column, select a dimension by selecting the check box next to the left of the dimension name.**

Note: 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.

9 **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.**

Note: If two independent dimensions have the same leaf member name for one varying attribute, deployment fails.

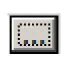
For example, suppose there is a varying attribute dimension, "VAYRPER," and two independent dimensions, "Period" and "Year." If the leaf member name is the same for "Period" and "Year," the deployment will fail.

To prevent this error, in at least one of the hierarchies representing an independent dimension, rename the dimension element that will be used as the leaf member . Then, retry the deployment.

10 **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.

11 **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 resize the **Independent Dimension Binding** dialog box.

12 **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 362

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.

13 **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.

14 Click **OK**.

15 **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 use, you may want to expand the **Independent Dimension Binding** dialog box.

16 Repeat Steps step 12 through step 15 as needed.

17 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 219.

The following topics describe the member properties you edit in the **Info** tab:

- "Selecting Consolidation Methods for Children Members" on page 225
- "Selecting the Member Calculation Solve Order" on page 227.
- "Selecting a Two Pass Calculation Option" on page 228
- "Selecting the Member Data Storage Method" on page 228
- "Selecting an Aggregate Storage Option" on page 230

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 226.

➤ 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 219.

2 Select the **Info** tab.

3 **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.

Note: For the member codes to use in external data sources, see “Using the Data Source to Work With Member Properties” and “Data Source Differences for Aggregate Storage Dimension Builds” in the *Oracle Essbase Database Administrator's Guide*.

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**.

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.

  Note: For the member codes to use in external data sources, see "Using the Data Source to Work With Member Properties" and "Data Source Differences for Aggregate Storage Dimension Builds" in the *Oracle Essbase Database Administrator's Guide*.

- 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 227.

➤ 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 219.
2. Select the **Info** tab.
3. Locate the **Solve Order** group.
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 206.

➤ 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 219.
2. Select the **Info** tab.
3. Locate 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 219****.**

2 **Select the Info tab.**

3 **Locate the Data Storage method group.**

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 229.

- **Existing**—Use the existing storage method.
- **External source**—In the drop-down list, select the column in the database that contains the storage method.

  Note: For the member codes to use in external data sources, see "Using the Data Source to Work With Member Properties" and "Data Source Differences for Aggregate Storage Dimension Builds" in the *Oracle Essbase Database Administrator's Guide*.

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 230.

➤ 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 219.

2 Select the **Info** tab.

3 Locate the **ASO Storage Options** group.

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.

Note: For the member codes to use in external data sources, see "Using the Data Source to Work With Member Properties" and "Data Source Differences for Aggregate Storage Dimension Builds" in the *Oracle Essbase Database Administrator's Guide*.

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 used for all member levels with the exception of member level 0 member.
- 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 219.

The following topics describe the member properties that you edit on the Account Info tab:

- "Selecting the Time Balance" on page 232
- "Selecting a Skip Option" on page 233
- "Selecting a Variance Reporting Method" on page 234
- "Specifying Data Load Scaling" on page 235

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 209.

➤ 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 219.
2. Select the Accounts dimension.
3. Select the member.
4. Select the **Account Info** tab.
5. Locate the **Time Balance** group.
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 210.

Note: For the member codes to use in external data sources, see "Using the Data Source to Work With Member Properties" and "Data Source Differences for Aggregate Storage Dimension Builds" in the *Oracle Essbase Database Administrator's Guide*.

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 219.

2 Select the Accounts dimension.

3 Select the member.

4 Select the **Account Info** tab.

5 Locate the **Skip** group.

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.

Note: For the member codes to use in external data sources, see "Using the Data Source to Work With Member Properties" and "Data Source Differences for Aggregate Storage Dimension Builds" in the *Oracle Essbase Database Administrator's Guide*.

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 219.

2 Select the Accounts dimension.

3 Select the member.

4 Select the **Account Info** tab.

5 Locate the **Variance Reporting** group.

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 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.
- **External source**—In the drop-down list, select the column in the database that contains the desired variance option.

Note: For the member codes to use in external data sources, see "Using the Data Source to Work With Member Properties" and "Data Source Differences for Aggregate Storage Dimension Builds" in the *Oracle Essbase Database Administrator's Guide*.

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 219.
2. Select the Accounts dimension.
3. Select the member.
4. Select the **Account Info** tab.
5. Locate the **Data load scaling** group.
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**.

Specifying Typed Measures

Typed measures extend the analytical capabilities of Essbase. In addition to numeric values, measures can also be associated with text- or date-typed values.

- Text measures are tagged as "text" in whichever dimension measures are represented. They enable cell values to contain one of an enumerated list of text labels. These labels are defined, at the outline level, using a mapping artifact called a Text List object.
- Date measures are tagged as "date" in the dimension where measures are represented. Date measures enable cell values in the form of a formatted date.

  If your Essbase model contains a date measure, be sure to review the information in "Deploying Cubes" on page 264 before you deployment.

The following general guidelines apply to both text and date measures:

- Add them to the existing measures dimension; for example, Accounts.
- Do not aggregate them. By default, text and date measures are assigned the non-aggregation symbol (^).

- Queries should be made at the same level at which data was loaded.

➤ To specify a typed measure:

1 **Ensure that the typed measure that you will use exists in the Metadata Navigator.**

2 Access the member properties tabs of the **Essbase Model Properties** dialog box using the procedure in “Accessing the Member Properties Tabs” on page 219.

3 Select the **Typed Measures** tab.

4 **Optional:** In the **Member format string** text box, enter an MDX expression to reformat the values in cells to be displayed in a desired format; for example, with a decimal point or a comma, or as text.

Note: Do not confuse the member string format with derived text measures. See “Creating or Editing Dimension Elements and Derived Text Measures” on page 132.

Note: The MDX expression cannot exceed 256 characters.

5 In the **Text List Association** group, navigate to a text list.

6 Select the text list.

The text list name appears in the **Current Assigned Text List** field.

7 Click **Apply.**

Adding Formulas to Members

A formula determines how Essbase calculates the relationships between members of an Essbase database.

For more information on formulas, see “About Formulas” on page 211.

For information on members, see “Members Overview” on page 219.

➤ 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 219.

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 column that contains the formula you want to use.

Note: The length of the formula cannot exceed 64 KB.

Note: For the member codes to use in external data sources, see "Using the Data Source to Work With Member Properties" and "Data Source Differences for Aggregate Storage Dimension Builds" in the *Oracle Essbase Database Administrator's Guide*.

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 237.

For information on members, see "Members Overview" on page 219.

The following topics describe the member properties you edit in the **Alias** tab:

- "Displaying Alias Sets" on page 238
- "Changing and Reformatting Alias Names" on page 239
- "Creating a Search Rule to Change and Reformat Alias Names" on page 239
- "Modifying a Search Rule to Change and Reformat Alias Names" on page 240
- "Changing Cases in Aliases" on page 241
- "Reformatting Spaces in Aliases" on page 241
- "Adding Prefixes to Aliases" on page 242
- "Adding Suffixes to Aliases" on page 243

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 237
- "Using Multiple Aliases" on page 238

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 147.

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:

Product	**Default**	**Long Names**
100-10	Cola	The Best Cola
100-20	Diet Cola	Diet Cola with Honey
100-30	Caffeine Free	Cola All the Cola, none of the Caffeine

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 "About Alias Sets" on page 147.

You create your alias sets using the Alias Set dialog box. See "Creating Alias Sets" on page 148.

Displaying Alias Sets

In the **Alias** tab, you can see a list of alias sets that are assigned to a member.

For information on aliases, see "Aliases Overview" on page 237.

For information on members, see "Members Overview" on page 219.

➤ 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 219.
2. Select the **Alias** tab.

   In the grid at the top of the dialog box, the alias sets assigned to the model are displayed in the **Alias Set** column.

   If the member selected in the left pane has an alias set mapped to it, the **Source Mapping** column shows the data source connection, table, and column of the alias mapping.

   Note: The alias section 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 239.
- See "Modifying a Search Rule to Change and Reformat Alias Names" on page 240.

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 237.

For information on members, see "Members Overview" on page 219.

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 219.
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 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 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 219.

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 237.

For information on members, see “Members Overview” on page 219.

➤ 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 219.
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 237.

For information on members, see “Members Overview” on page 219.

➤ 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 219.
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 243.

See "Aliases Overview" on page 237.

See "Members Overview" on page 219.

➤ 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 219.

2 Select the **Alias** tab.

3 Select a prefix option from the drop-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 237.

- **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** drop-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 242

See "Aliases Overview" on page 237.

See "Members Overview" on page 219.

➤ 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 219.**

2 **Select the Alias tab.**

3 **Select a suffix option from the pull-down list:**

- **None**—No suffix will be added to the alias

  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 237.

- **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** drop-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 214.

See "Members Overview" on page 219.

➤ 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 219.
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.

   Note: For the member codes to use in external data sources, see "Using the Data Source to Work With Member Properties" and "Data Source Differences for Aggregate Storage Dimension Builds" in the *Oracle Essbase Database Administrator's Guide*.
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 237.

For information on members, see "Members Overview" on page 219.

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 245.

The following topics describe the member properties you edit in the **Transformations** tab:

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 246.

For information on members, see "Members Overview" on page 219.

➤ 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 219.

2 Select the **Transformations** tab.

3 Select a **Prefix** option from the drop-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** drop-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 245.

For information on members, see "Members Overview" on page 219.

➤ 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 219.
2. Select the **Transformations** tab.
3. Select a **Suffix** option from the drop-down list:
   - **None**—No suffix will be added to the member.

     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 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: Selecting **All ancestors' names** for a suffix may produce a lengthy suffix. 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 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** drop-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 248.
- See "Modifying a Search Rule to Change and Reformat Member Names" on page 249.

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 219.

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 245.

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 219.
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 219.

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 219.

➤ To change the case of a members:

1 Access the member properties tabs of the **Essbase Model Properties** dialog box using the procedure in “Accessing the Member Properties Tabs” on page 219.

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 219.

➤ 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 219.

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 251
- "Validating Model Properties" on page 253
- "Reviewing Changes to Properties" on page 253
- "Browsing Models" on page 254
- "Using Tool Tips" on page 255

See "Essbase Models Overview" on page 184.

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 "Essbase Models Overview" on page 184.

The following topics discuss the tasks you can perform in the model work area:

- "Opening the Model Work Area" on page 251
- "Using the Model Work Area" on page 252

Opening the Model Work Area

You can use the model work area to see a graphical representation of dimensions and members.

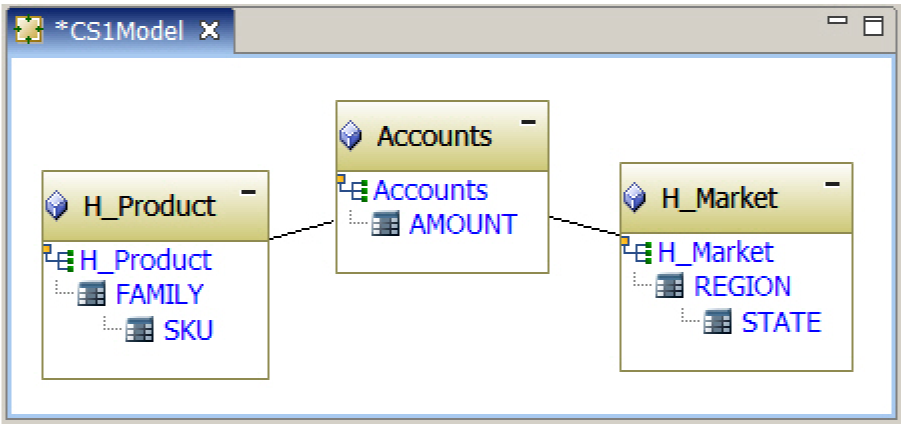
➤ To open the model work area:

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:

Figure 16 Model Work Area



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 264.
- **Edit properties**—To edit model, dimension, and member properties, right-click in the model work area and select **Essbase Properties**.

  Alternatively, you may also right-click on dimensions or member names in the

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 "Essbase Models Overview" on page 184.

➤ To validate a model:

1. Access the **Essbase Model Properties** dialog box using the procedure in "Accessing the Essbase Model Properties Dialog Box" on page 185.
2. Click the **Run validation** button,  , 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,  , 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 "Essbase Models Overview" on page 184.

➤ To see a list of changes to the model properties:

1. Access the **Essbase Model Properties** dialog box using the procedure in "Accessing the Essbase Model Properties Dialog Box" on page 185.

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 "Essbase Models Overview" on page 184.

➤ 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 Essbase Model Properties Dialog Box" on page 185****.**

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. **Optional: After editing or viewing the properties of several dimensions and members, to return to the 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. **Optional: 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 **Optional:** 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 Essbase Model Properties Dialog Box" on page 185.

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.

12

Model Resync

In This Chapter

About Model Resync

If you change either of these metadata elements, **cube schema** or **hierarchy**, you can use Model Resync to propagate the changes to any Essbase model that uses the metadata element. With Model Resync, you do not have to recreate models or re-configure any settings.

When you change a metadata element, Essbase Studio tags any model which uses the element as being out of sync with the latest version of the element. You can then choose to resync the model.

Events that trigger the need for a model resync include:

- Cube schema changes:
  - Adding a hierarchy to the cube schema
  - Removing or replacing a hierarchy
  - Moving a hierarchy
  - Modifying a hierarchy
  - Renaming the schema
- Hierarchy changes:
  - Adding a chain to the hierarchy
  - Removing or replacing a chain
  - Moving a chain
  - Modifying a chain
  - Renaming the hierarchy

When you start to change a metadata element, you will see a warning if the element is used in any Essbase models. You will also see a listing of all models which use the element. If you decide to continue the update, all models that become out of sync are denoted by this icon next to the model name in the **Metadata Navigator**: .

Note: If you resync the model, the settings of the other, unchanged elements will retain their current values. All chains that are unchanged in a hierarchy will also retain their current settings. If any elements are renamed, all other elements will retain their current name.

Using Model Resync

After you have updated a metadata element, use Model Resync to propagate the changes to the Essbase models which use that metadata element.

There are various types of model resyncing:

- You can resync the models affected by changes to a specific metadata element. See "Metadata Element-Model Resync" on page 258.
- You can resync one model, basing the resync on multiple elements. See "Model Resync" on page 258.

Metadata Element-Model Resync

➤ To resync those models affected by changes to a specific metadata element:

1. **In the Metadata Navigator, highlight the updated metadata element.**
2. **Right-click, then select Update Out-of-sync Models**

   The **Sync Models** dialog box is displayed. All models needing to be resynced by changes to the element are listed in **Out-of-sync models**.
3. **Select the models to update and move them to Models to sync.**

   Note: When a model is resynced using this method, it is updated to reflect changes only to this specific element. No other elements in the model are updated.
4. **Click OK to sync the models with the updated metadata elements.**

   Note: The model may still be out of sync if there are other elements which have been changed. Also, the model may be invalid after the resyncing operation. You should re-validate all models that have been resynced.

Model Resync

➤ To resync one model, basing the resync on multiple elements:

1. **In the Metadata Navigator, highlight the out-of-sync model.**

   All out-of-sync models are denoted by this icon next to the model name in the **Metadata Navigator**:

2 **Right-click, then select Update Out-of-sync Model**

The **Sync Models** dialog box is displayed. All updated metadata elements are listed in **Modified hierarchies and cube schema**.

3 **Select the updated elements you wish to sync with the model and move them to Objects to sync with model.**

Note: When a model is resynced using this method, it is updated to reflect changes only to the elements you specify. No other elements in the model are updated.

4 **Click OK to perform the partial resyncing.**

Note: The model may still be out of sync if there are other elements which have been changed. Also, the model may be invalid after the resyncing operation. You should re-validate all models that have been resynced.

13 Cube Deployment

In This Chapter

About Cube Deployment

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 when you deploy a cube:

- Build outline only
- Load data only
- Build outline and load data
- Load members incrementally
- Delete members from a cube before loading
- Delete and restore data to a cube
- Deploy in streaming or non-streaming mode
- Create and save an Essbase rules file, which you can use to load members or data or both at a later time
- Create and save a MaxL load script containing all the cube deployment options you have selected, which you can use to load members or data or both at a later time

You can adjust how data is retrieved during a cube deployment depending on these factors:

- How you set the `server.essbase.streamingCubeBuilding` server property
- The options you choose in the **Data source settings** group in the **Cube Deployment Wizard**
- Whether you choose the streaming mode deployment option.

See "Deployment Scenarios and Streaming Cube Building Property Considerations" on page 262 and "Providing Connection Information for Cube Deployment" on page 266.

If you used a date measure in your cube schema, see "Deploying Date Measures" on page 264 for information on performing cube deployment so that data is aggregated correctly.

If you did not choose to create an Essbase model during cube schema creation, you can create one at any time. You can also create more than one Essbase model for the same cube schema, as described in "Creating Essbase Models from Existing Cube Schemas" on page 181. After creating the Essbase model, launch the **Cube Deployment Wizard** and complete the first page, as described in "Providing Connection Information for Cube Deployment" on page 266. From the wizard, you can also modify Essbase properties for the selected 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. An option in the **Cube Deployment Wizard** creates the deploy script automatically. All the options that are available in the **Cube Deployment Wizard** are also available in the deploy command and are recorded in the MaxL script. See the MaxL documentation in the *Oracle Essbase Technical Reference* for information.

Deployment Scenarios and Streaming Cube Building Property Considerations

For cube deployment, Essbase Studio provides three data-retrieval mode options:

- Nonstreaming mode, with the connection string derived by Essbase Studio Server (the default)
- Nonstreaming mode, with user-provided ODBC DSN
- Streaming mode, with the connection string derived by Essbase Studio Server

Use of these modes depends on how you have set the streaming cube building property, `server.essbase.streamingCubeBuilding`, as described in this topic.

For information about setting the `server.essbase.streamingCubeBuilding` property, see "server.essbase.streamingCubeBuilding" on page 34.

Nonstreaming mode, with the connection string derived by Essbase Studio Server (the default)

`server.essbase.streamingCubeBuilding=false`

In the **Cube Deployment Wizard**, select **ODBC (Essbase dynamically creates ODBC connection string)**.

This option allows Essbase to conveniently establish connection to a data source during deployment and assumes default values for all connection attributes. In this case, Essbase Studio uses ODBC drivers during deployment.

Nonstreaming mode, with user-provided ODBC DSN

`server.essbase.streamingCubeBuilding=false`

In the **Cube Deployment Wizard**, select **ODBC (Enter ODBC DSN name)**, and enter an ODBC DSN that you have previously set up.

This option allows Essbase to take advantage of the attributes specified in an ODBC DSN. For example, if the Essbase model that you are deploying is based on a Unicode data source, you can create an ODBC DSN with "N-CHAR Support" enabled, and then specify this DSN at deployment time.

Streaming mode, with the connection string derived by Essbase Studio Server

`server.essbase.streamingCubeBuilding=true`

In the **Cube Deployment Wizard**, select "ODBC (Essbase dynamically creates ODBC connection string)."

This option allows Essbase to conveniently establish connection to a data source during deployment and assumes default values for all connection attributes. In this case, Essbase Studio uses JDBC drivers during the deployment.

The following table provides guidelines on deployment modes and possible deployment scenarios.

Table 5 Guidelines on Deployment Modes and Possible Deployment Scenarios

Deployment Mode	Deployment Scenario Examples
Nonstreaming with connection string (default) `server.essbase.streamingCubeBuilding=false`	When the data source is relational, and you have no special requirements to adjust the default ODBC DSN connection attribute values
Nonstreaming with ODBC DSN `server.essbase.streamingCubeBuilding=false`	When you want to change default values of connection attributes. Examples: ● When an Essbase model is based on Unicode source database, "N-CHAR support" must be enabled ● If you want to enable data compression to improve data transmission performance, "Wire Protocol Mode" can be set to `2` ● An encryption option is selected as an authentication method for IBM DB2 data source connections ● On Windows, when Essbase is started using OPMN, for cubes built from Oracle BI EE data sources

Deployment Mode	Deployment Scenario Examples
Streaming `server.essbase.streamingCubeBuilding=true`	Examples: ● Essbase model is built from an Performance Management Architect data source ● Essbase model is built and deployed to Essbase from Performance Management Architect ● Essbase model based on Unicode source database (N-CHAR support) ● Authentication Method of an IBM DB2 data source connection is an encryption option (Encrypt Password, Encrypt UserID and Password, or Client) ● On Windows, when Essbase is started using OPMN, for cubes built from Oracle BI EE data sources ● When the Essbase model being deployed contains a calendar hierarchy

Deploying Date Measures

If you plan to deploy cubes that contain a date measure, you should know that Essbase Studio applies the MAX aggregation function to the date measure column when generating data load SQL so that the result set will come out to the last transaction date.

In the following example, the Date column is used as a date measure:

```
Product     State     Sales      Date
-------    -----     ------     ----------
100-10      CA        200.00     2009-01-31
100-10      CA        500.00     2009-12-31
```

Using the MAX aggregation function, data is aggregated on the 2009-12-31 date, and Sales totals 700.00.

Oracle suggests that you perform the following tasks when deploying a cube that contains a date measure:

- In the **General** tab of **Essbase Model Properties**, select a date format.

  The default date format is `yyyy-mm-dd`.
- In the **Cube deployment options** page of the **Cube Deployment Wizard**, be sure to select **Overwrite existing data** from the **Load data options** group.

Deploying Cubes

The **Cube Deployment Wizard** consists of these screens:

- **Essbase Server connection options**—You provide information regarding the Essbase Server connection and the method for connecting and retrieving data from the data source. See "Providing Connection Information for Cube Deployment" on page 266.

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 an Essbase Server Connection" on page 265. 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 271.
- **Cube deployment options**—Set load options, specify rejected records handling, and schedule immediate deployment or choose to save the deployment information you have entered as a MaxL script. See "Setting Deployment Options" on page 268.

You can also review the history of cube deployments for each Essbase model, as described in "Viewing Deployment History" on page 275.

Creating an Essbase Server Connection

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.**

   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:**
   - A **Name** for this connection
   - An optional **Description**.
   - The **Server** name where the Essbase instance or cluster resides, and the **Port** number.

     The default for **Server** and **Port** is the Essbase instance or cluster running on the same machine as Essbase Studio.

     Note: Cluster names are case sensitive. The default cluster name is `EssbaseCluster-1`.
   - The name of the **User** and the **Password** needed to access this Essbase Server.
   - A **Data Encryption** method:
     - **No Encryption**—Thhe default mode for communication between Essbase Studio Server and an Essbase Server instance or cluster.

- ❍ **SSL**—If the Essbase Server instance or cluster to which you are connecting is configured for Secure Socket Layer (SSL) protocol over TCP/IP, select this method.

  SSL enables secure communication between Essbase Studio Server and Essbase.

  Note: If the Essbase Server is not configured to support SSL, then a connection cannot be established and an error message is displayed.

- If you are connecting to an Essbase Server cluster, select **Cluster**.

Note: Essbase Server clusters must be set up prior to creating an Essbase connection. See *Oracle Hyperion Enterprise Performance Management System Installation and Configuration Guide* and *Oracle Hyperion Enterprise Performance Management System High Availability Guide* for information on setting up clusters.

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 Information for Cube Deployment

Use the “Essbase Server connection options” page of the **Cube Deployment Wizard** to specify Essbase and data source connection information. You can also modify Essbase model properties before deployment.

➤ To set up server, data source, 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.**

   The **Essbase Server connection options** page is launched.
3. **In Essbase Server Connection, select the connection representing the Essbase Server to which you want to deploy.**

   To deploy to an Essbase Server connection that is not on the list, click **New Connection** and complete the steps in “Creating an Essbase Server Connection” on page 265, then proceed to 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, Oracle strongly recommends that you use a new application and database name, or use the **Delete all members first** option to deploy over an existing XOLAP application.
5. **Optional: Select the Enable streaming mode for cube deployment check box to perform this cube deployment in streaming mode.**

Note: The **Enable streaming mode for cube deployment** check box setting is not retained across deployments. To use the check box for streaming mode, it must be set each time you perform a cube deployment.

Streaming mode means that during cube deployment, Essbase Studio Server queries the external data source directly (rather than querying the external data source using an ODBC connection).

The **Enable streaming mode for cube deployment** check box is enabled when the Essbase model being deployed contains single or multiple relational data sources.

The check box is disabled when the data sources used in the Essbase model are one or more text file sources, one or more Dimension Server (Performance Management Architect) sources, or a mix of text file and relational sources.

If Essbase Studio Server is running in nonstreaming mode (the `server.essbase.streamingCubeBuilding` property is set to false), selecting this option will override that setting.

6 **In the Data source setting group, select an option:**

- **ODBC (Essbase dynamically creates ODBC connection string)**—Use the data source name dynamically created by Essbase. This is the default.
- **ODBC (Enter ODBC DSN name)**—Enter the ODBC DSN name to use.

  If you choose to deploy using an ODBN DSN name in order to take advantage of your own custom ODBC DSN parameter settings, follow these guidelines:

  - Set up your ODBC DSN before beginning deployment, on the server machine where Essbase is installed.
  - The ODBC DSN must have the same user name and password as the data source connection being used in this deployment.

  Note: For additional information on using the ODBC settings in conjunction with the **server.essbase.streamingCubeBuilding** property, see "Deployment Scenarios and Streaming Cube Building Property Considerations" on page 262.

- **OCI (Enter OCI connect identifier)**—Enter the Oracle OCI connect identifier to use.

  The syntax for an Oracle OCI connect identifier is:

  `host:port/SID`

  Following is an example OCI connect identifier where the host server name is "`myserver`," the port number is `1521`, and the Oracle SID (Service Identifier) is "`orcl`":

  `myserver:1521/orcl`

  Note: Other connect identifier formats can be used. Refer to Oracle documentation for more information on OCI.

By calling an ODBC DSN or OCI connect identifier, you can take advantage of the particular parameters that you have set in the DSN or in OCI. For example, during cube deployment, you may use an Oracle Wire Protocol driver that is set up to take advantage of driver performance and failover options.

All items in the **Data Source Settings** group are disabled under the following conditions:

- The **Enable streaming mode for cube deployment** check box is selected (step 5).
- Essbase Studio Server is running in streaming mode (the `server.essbase.streamingCubeBuilding` property is set to true).
- The Essbase model contains elements from multiple data sources
- The Essbase model was created using a text file data source
- The Essbase model was created using a Dimension Server (Performance Management Architect) data source

7 **Optional:** Click **Model Properties** if you want to modify the Essbase properties of the model.

See Chapter 11, "Essbase Properties.".

8 Click **Next** to launch the **Cube deployment options** page of the wizard.

Note: If errors exist in the model, you are warned when you click **Next**. A dialog is displayed asking if you want to launch the **Essbase Model Properties** dialog box. Click "**Yes**" to launch the properties dialog box and correct the errors.

Setting Deployment Options

In the "Cube deployment options" page of the **Cube Deployment Wizard**, you specify load information and rejected records settings. You can also choose to save the deployment information in a MaxL script for deployment at another time.

➤ To specify deployment options:

1 In the **Cube deployment options** page of the **Cube Deployment Wizard**, choose a load option from the **Load task type** group:

- **Build outline**—Adds 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 dimensions and members without data to an Essbase outline and populates an Essbase database with data.

2 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.

Note: These options are enabled only if you chose a data load option in step 1.

3 **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 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 reason to do so. You should delete all members if, for example, you know that some members have been removed from an Essbase model and you want to build a smaller Essbase outline that contains the smaller set of members.

Note: This check box is not enabled when you choose to only load data.

4 **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 time required for the load.

Note: This check box is not enabled when you choose to load only data.

5 **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. It is also not enabled if, when building an outline, you select the **Delete all members first** option or the **Delete and restore database** option.

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 271.

6 **Optional: Select Create and save rule file only to use later to load members and data from Administration Services Console.**

Rules files generated by Essbase Studio contain SQL statements that specify the changes Essbase should make to members and data 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. An Essbase Studio-generated rules file contains SQL statements that describe which build method to use, whether data values or members are sorted or are in random order, and how to transform data values or members before loading them. Essbase Studio creates a separate rules file for each dimension; however, if a measures dimension exceeds 250 members, then Essbase Studio creates one rules file for every 250 members.

Rules files are saved to the `app` directory of your Essbase installation.

For more information, see "Rules File Limitations and Guidelines" on page 364.

7 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:

  EPM_ORACLE_HOME/products/Essbase/EssbaseStudio/server/ess_japihome/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 `ess_japihome/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`

c. **Optional:** Select **Stop the data load when the first record is rejected** to stop a data load after Essbase 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.

8 In **Scheduling Options**, perform an action:

- To deploy the cube immediately, verify that **Deploy now** is selected.

  The "Deploy now" option is selected by default.

- To create a MaxL load script containing the deployment information, select **Save as MaxL load script**, click **Browse** and navigate to the location where the MaxL script will be saved, and then click **Save**.

Note: At least one option from the **Scheduling Options** group must be selected.

9 **Perform an action:**

- If you selected the **Incremental Load** option, click **Next** and complete the tasks in "Setting Up an Incremental Load for Cube Deployment" on page 271.
- Click **Finish** to deploy the cube.

  The **Deployment Succeeded** dialog box is displayed for successful deployments, containing status information about the deployment.

  Note: If you encounter an out-of-memory error during cube deployment, increase the virtual memory setting for Essbase Studio Console, as described in "Configuring Virtual Memory" on page 62.

Deployed cubes are displayed in the **Metadata Navigator** as children of the Essbase model from which they were deployed.

After deployment, view and edit the outline in Administration Services Console. You can also query the cube using Smart View or Essbase Spreadsheet Add-in.

If you chose to save the deployment information to a MaxL script, note that shell variables, $1 and $2, are used for the user name and password parameters respectively in the script.

For example:

```
deploy all from model 'cs1Model' in cube schema '\CubeSchemas\cs1' login $1
identified by $2 on host 'poplar-pc1' to application 'cs2' database 'cs2' add
values using connection 'Connection1' keep 200 errors on error ignore dataload
write to default;
```

In the above example, when using this script, you would make the following substitutions at the command line for Essbase login information:

- `$1` = user name
- `$2` = password

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 an option:**

- **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 a Preserve option:**

- **All Existing Data**—This is the default. Preserves all existing data that applies to the changed outline when restructuring occurs.
- **Input data**—Preserves only 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 must 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, reducing the disk space for restructuring and improving 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 **Perform an action:**

- Click **Finish** to deploy the cube.

  The **Deployment Succeeded** dialog box is displayed for successful deployments, containing details about the deployment.

  Deployed cubes are displayed in the **Metadata Navigator** as children of the Essbase model from which they were deployed.

  You can now view and edit the outline in Administration Services Console. You can also query the cube using Smart View or Essbase Spreadsheet Add-in.

- If you want to save the deployment as a MaxL load script, click **Next** and complete the tasks in "Setting Deployment Options" on page 268.

Viewing Deployment Progress and Results

When you launch a cube deployment, a window is displayed that provides you with deployment information and statistics.

The window displays information on the progress of the deployment. When the deployment is complete, the window displays the following information:

- Deployment start and completion time
- Elapsed deployment time
- Deployment runtime statistics:
  - Number of records processed
  - Number of records rejected, if any
- The error file location (if there are rejected records)
- A list of the rejected records, if any

Notes

The information and statistics displayed depend on whether Essbase Studio Server is running in streaming or nonstreaming mode.

- If Essbase Studio Server is run in nonstreaming mode, deployment runtime statistics are displayed onscreen for both processed records and, if they exist, rejected records.

  For example, after deployment, you may see onscreen:

  ```
  Records processed 17. records rejected: 3.
  ```

  In nonstreaming mode, the server property, `server.essbase.streamingCubeBuilding`, is set to `false`.

- If Essbase Studio Server is run in streaming mode, the deployment runtime statistic is displayed onscreen only for processed records. The rejected records statistic, if any exist, is not displayed onscreen. After deployment is finished, the error file contains the list of rejected records.

  For example, after deployment, you may see onscreen:

  ```
  Records processed 17.
  ```

  In streaming mode, the server property, `server.essbase.streamingCubeBuilding`, is set to `true`.

See "server.essbase.streamingCubeBuilding" on page 34 for more information on this property.

14 Deployed Applications

In This Chapter

Viewing Deployment History

You can view the deployment history of Essbase models 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 an Essbase model.
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.

     In the case of a rehosted Essbase Server where the “Replicate” option was selected, a copy of the last successful deployment history is listed containing the original date and time of the deployment, along with an updated copy containing the date and time of the rehosting.
   - **Essbase Server**—The Essbase Server instance name to which the cube was deployed.

     In the case of a rehosted Essbase Server where the “Replicate” option was selected, a copy of the last successful deployment history is listed containing the original host:port information, along with an updated copy containing the new host:port information.
   - **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 name of the user who deployed the cube.

3 **Double-click a row in the grid representing a deployment to view the Deployment Errors and Warnings dialog box.**

Errors or warnings logged during deployment are listed.

4 **Click OK or Cancel to close Deployment Errors and Warnings.**

5 **When finished, click the X in the tab of the Deployment History window to close it.**

Updating Cube Linkage

This topic contains these subtopics:

- "About Cube Linkage" on page 276
- "Before Updating Cube Linkage for Individual Cubes" on page 277
- "Updating Cube Linkage for Individual Cubes" on page 277
- "Updating Cube Linkage After Essbase Studio Rehosting" on page 278

About Cube Linkage

Cube linkage refers to the information stored in Essbase cubes regarding:

- The Essbase Studio Server and port used to deploy the cube
- The model on which the cube was based
- The login information for the data source from which the cube was built

For cubes built using Essbase Studio, you can update cube linkage information to change the Essbase Studio Server, port, and model context.

Examples of using cube linkage:

- When drill-through reports are redirected to a new Essbase Studio Server or are based on a different Essbase model. This can be a useful load balancing strategy.
- When data source login information changes, updating cube linkage is especially helpful for XOLAP-enabled cubes. By updating the cube linkage, data source login information is updated without recreating the model or redeploying the cube.

  To update cube linkage in the types of cases above, complete the procedures in "Before Updating Cube Linkage for Individual Cubes" on page 277 and "Updating Cube Linkage for Individual Cubes" on page 277.
- When Essbase Studio Server is rehosted on a different machine (for example, after upgrade), the process of updating references to the rehosted Essbase Studio Server is handled using cube linkage.

  If you have rehosted Essbase Studio Server, follow the procedure in "Updating Cube Linkage After Essbase Studio Rehosting" on page 278.

Note: Always perform the rehosting steps for Essbase before performing the steps in "Updating Cube Linkage After Essbase Studio Rehosting" on page 278. See "Updating References to a Rehosted Essbase Server" on page 63 for more information.

See the *Oracle Hyperion Enterprise Performance Management System Installation and Configuration Guide* for complete information on EPM System product rehosting.

Before Updating Cube Linkage for Individual Cubes

➤ Before completing any cube linkage update procedure, perform the following actions:

1. **Verify the Essbase Studio Server machine name and port number information.**
2. **Ensure that the associated catalog for any Essbase Studio Server instances that you enter contains the Essbase model names that you want to link to.**
3. **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 99.**

Updating Cube Linkage for Individual Cubes

Before beginning this procedure, complete the steps in "Before Updating Cube Linkage for Individual Cubes" on page 277.

➤ To update the cube linkage for all cubes built using Essbase Studio:

1. **In Essbase Studio Console, select Tools, and then Update Cube Linkage.**
2. **In Update Cube Linkage, ensure that Update selected Essbase application and database is selected.**

   This default selection applies the new Essbase Studio Server that you specify (in the subsequent steps of this procedure) only to the selected Essbase cube.

3. **Under Essbase Connections, expand the applicable Essbase connection and application, and then select the Essbase cube whose linkages you want to change.**

   You can update linkage only for cubes built by Essbase Studio. Cubes that have entries under **Cube Linkage Essbase Studio Server** and **Cube Linkage Essbase Model** are built by Essbase Studio and can be updated. Cubes with blank entries in those columns cannot be updated.

4. **Click Update to launch the Cube Linkage dialog box.**
5. **Enter the new Essbase Studio Server name and port number in the following format:**

   `essbase_studio_server_name:port_number`

   For example:

   `aspen3:5300`

Tip: Essbase Studio remembers previously entered server name and port number combinations. If a desired combination is available in the drop-down list, you may select it.

Note: Essbase Studio does not validate the server name and port number combinations that you enter. No error message is displayed if you enter this information incorrectly, and you will not see the error until you perform a cube deployment.

Note: If you are updating cube linkage only because login information for the data source changed, you need not enter a new Essbase Studio Server name or port. Enter a new server name or port only to point to a new server or port.

6. Select an **Essbase model name** from the list.

   You can also manually enter an Essbase model name. Ensure that the Essbase model name that you enter exists in the catalog associated with the Essbase Studio Server entered in step 5.

   Note: Essbase Studio does not validate model names that you enter. No error message is displayed if you enter this information incorrectly, and you will not see the error until you perform a cube deployment.

   Note: If you are updating cube linkage only because login information for the data source changed, you need not enter a new model name. Enter a new model name only to point to a new model.

7. Click **OK** to return to **Update Cube Linkage.**
8. For the selected cube, verify the entries under **Cube Linkage Essbase Studio Server** and **Cube Linkage Essbase Model.**
9. Repeat step 3 through step 8 for all cubes that require updates.
10. To exit, click **Close.**

Updating Cube Linkage After Essbase Studio Rehosting

If you have rehosted Essbase Studio Server (for example,after upgrade), follow the procedure in this topic.

See the *Oracle Hyperion Enterprise Performance Management System Installation and Configuration Guide* for complete information on EPM System product rehosting.

Note: Always perform the rehosting steps for Essbase before performing the steps in this procedure. See "Updating References to a Rehosted Essbase Server" on page 63 for more information.

➤ After Essbase Studio rehosting, to update the cube linkage for cubes built using Essbase Studio:

1 **In Essbase Studio Console, select Tools, and then Update Cube Linkage.**

2 **Select Update all Essbase applications and databases to link to the current Essbase Studio Server (all Essbase instances must be started).**

All Essbase applications and databases will be linked to the Essbase Studio Server to which you are currently connected.

Note: All Essbase instances or clusters must be running for the cube linkage update to take effect.

3 **Optional: To keep all Essbase applications running after the update, clear the Stop all Essbase applications after the update check box.**

The default is to stop all Essbase applications after update.

4 **Click Update.**

A message indicates that the cube linkage update is successful.

If unsuccessful, an error message shows the details.

5 **To exit, click Close.**

15 Lineage

In This Chapter

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.

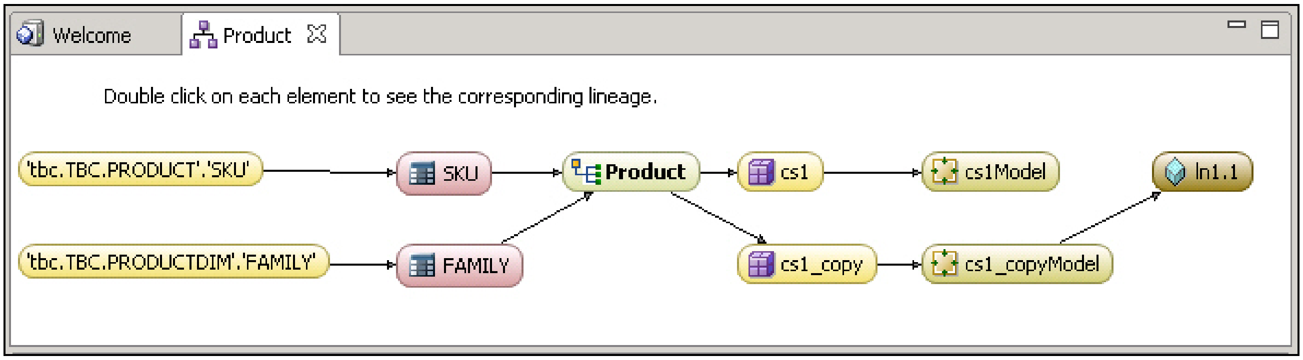
The lineage work area enables you to view information ranging from which data source table and column the element comes from to the deployed cubes that the element is used in. The lineage work area can help you quickly understand the potential impact of changes to the element; for example, the effect 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 that you have chosen. For example, a standard hierarchy would show these related objects:

- The hierarchy columns, including the caption and key bindings for each column on which the hierarchy was built
- 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)

Figure 17 shows an example lineage diagram.

Figure 17 Lineage for a hierarchy named “Product”



Lineage can also show you when the caption and key bindings for a dimension element differ. Figure 18 shows a dimension element with different caption and key bindings.

Figure 18 Lineage where key binding and caption binding are different

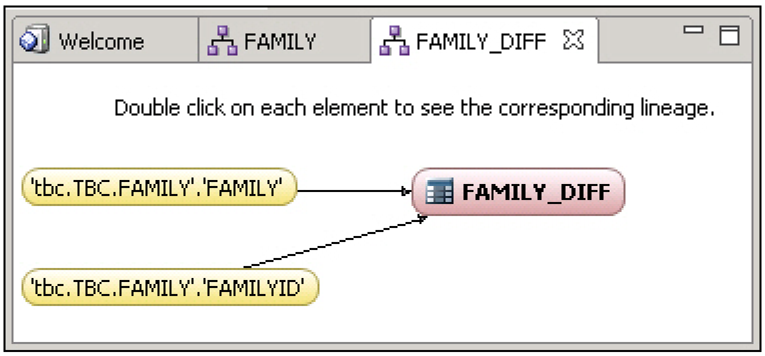
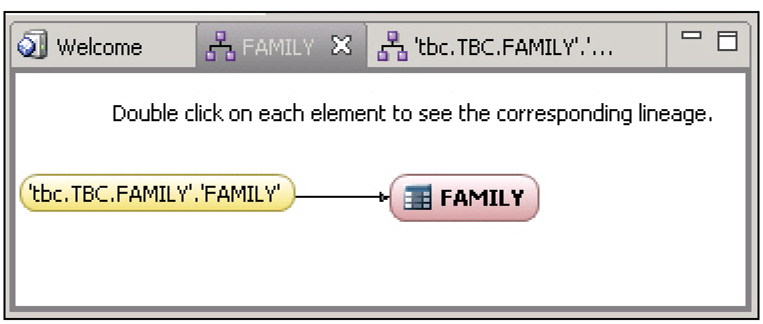


Figure 19 shows a dimension element where either the caption and key bindings are the same or the key binding expression for the element is set to **Delayed**.

Figure 19 Lineage where key binding and caption binding are the same or key binding is set to “Delayed”



Note that the illustration in Figure 19 can also mean that the key binding for the element is “delayed”. If you right-click the FAMILY column and the pop-up menu displays only the option, **View Sample With Caption Binding**, then the key binding expression for the element is set to **Delayed**. If you right-click and see only the **View Sample Data option**, then the caption and key binding expressions are the same.

To display the lineage of an element, see “Opening the Lineage Work Area” on page 283.

For information on optional tasks that you can perform in the lineage work area, see “Using the Lineage Work Area” on page 283.

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, in the **Metadata Navigator**, right-click a metadata element and select **Show Lineage**.

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 283

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**.
- **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 an Essbase model, right-click in the model and select **Show Deployment History**.
- **View Sample Data**—Depending on how the selected element is constructed, different sample viewing options are available. To view the sample data for an element, right-click on the element.

  If the caption binding and key binding expressions for the selected element are the same, only **View Sample Data** is displayed.

  If the caption binding and key binding expressions for the selected element are different, both the **View Sample With Caption Binding** and the **View Sample With Key Binding** menu items are displayed.

  If the key binding expression for the element is **Delayed**, then only the **View Sample With Caption Binding** menu item is displayed.

16 Drill-through Reports

In This Chapter

Drill-through Reports Overview

With drill-through reports, you create spreadsheet reports that display data retrieved directly from external relational 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 and summarize the data that spreadsheet users most likely need. 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 10 lowest performing stores in California?
- What are the addresses of the stores?
- What are the names of the store managers?

Such detailed information is not usually stored in an Essbase database, but 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 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 364.

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 Oracle Hyperion Financial Data Quality Management, Fusion Edition and Oracle BI EE 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 User's Guide Help*.
- Using drill-through reports with Spreadsheet Add-in, see the *Oracle Essbase Spreadsheet Add-in User's Guide*.

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.

The steps to defining a drill-through report are:

- "Creating a Drill-through Report" on page 287.
- "Specifying Report Intersection Levels" on page 287.
- "Defining the Report Type and Customizing the Report" on page 289.
- "Associating Drill-through Reports with Essbase Models" on page 306.

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, and then Drill-through Report.**

   If you select a folder and then right-click and select the Drill-through Report option, the drill-through report is stored in that folder; if you right-click on Root, it is stored at the root level.

2 **Enter a Name.**

   Drill-through report names cannot exceed 50 characters.

   See "Drill-through Reports Limitations and Guidelines" on page 364.

3 **Optional: Enter a Description.**

4 **Complete these tasks:**

   - Specify the report intersection levels; see "Specifying Report Intersection Levels" on page 287.
   - Define the report type and customize the report; see "Defining the Report Type and Customizing the Report" on page 289.
   - Associate the report with one or more exported models; see "Associating Drill-through Reports with Essbase Models" on page 306.

Specifying Report Intersection Levels

In the Context tab of the drill-through report editor, when you define intersection levels for a drill-through report, you set the common thread that ties this drill-through report to other drill-through targets.

See "Determining Where to Set an Intersection Level" on page 288.

➤ To specify report intersection levels:

1 **In the Context tab of the drill-through report editor, click Add.**

The **Select Hierarchy** dialog box is displayed.

2 **In Select Hierarchy, navigate to a hierarchy, measure hierarchy, or recursive 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.

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 need not click the **Add** button.

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.

3 **In the expanded hierarchy under Intersections, select the check box next to the member name that 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 step 2 through step 3 to add hierarchies for all drill-through intersections that you want to define.**

Note: In Smart View only: If you select only one hierarchy, 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 **If you selected recursive hierarchies for this drill-through report and want to modify the default intersection, which is level 0, click Advanced Settings to open the Recursive Hierarchy Settings dialog box.**

Note: If a recursive hierarchy is also a measure hierarchy, it is treated as a regular hierarchy in the drill-through report editor, and the **Advanced Settings** button is not enabled.

See “Specifying Intersection Levels for Recursive Hierarchies” on page 289 for information.

6 **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 289.**

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 built 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 ensures 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.

Specifying Intersection Levels for Recursive Hierarchies

You can create a drill-through intersection on a recursive hierarchy, specifying the generation or level that will participate in the report. The drill-through report can be viewed in Smart View or Spreadsheet Add-in.

For recursive hierarchies, the default drill-through intersection is level 0.

See "Recursive Hierarchies" on page 164 for information on recursive hierarchies.

➤ To specify the drill-through intersection generation or level for a recursive hierarchy:

1. Complete step 1 through step 5 in "Specifying Report Intersection Levels" on page 287.
2. In **Recursive Hierarchy Settings**, under **Generation/Level**, click in the cell for the appropriate recursive hierarchy row and specify whether the intersection for this drill-through report will be based on Generation or Level.

   The default setting is Level.
3. Under **Number**, specify the generation or level number to use as the drill-through intersection.

   The default setting is `0` (zero).
4. Repeat step 2 through step 3 for all recursive hierarchies that are included in the drill-through report.
5. Click **OK** to save the recursive hierarchy settings that you specified and return to the **Context** tab for the drill-through report editor.
6. 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 289.

Defining the Report Type and Customizing the Report

In the **Report Contents** tab, you define the type of report that you want to create. The tasks you must complete depend on the type of report you define.

The report types that you can choose:

- **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 290.

- **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 298.

- **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 303.

Defining and Customizing a Report for a Relational Source

Use the **Report Contents** tab to perform these tasks:

- Define the report type and specify the drill-through report columns

  See "Defining the Relational Report Type and Specifying Drill-through Report Columns" on page 290.

- Define the sort order for drill-through report columns

  See "Defining Sort Order for Drill-through Report Columns" on page 292

- Define Template SQL

  See "Defining Template SQL" on page 293.

- Specify row governors

  See "Specifying Row Governors for Relational Sources" on page 294.

- Specify filters

  See "Specifying Drill-through Report Filters for Relational Sources" on page 294.

- Test the report

  See "Testing Reports for Relational Sources" on page 295 and "Example Testing Scenarios —When Caption and Key Bindings Differ" on page 296.

Defining the Relational Report Type and Specifying Drill-through Report Columns

➤ To define a relational drill-through report type and specify the report columns and their order:

1. Select the **Report Contents** tab of the drill-through report editor.
2. From **Drill-through Report Type**, select **Relational.**
3. 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 3.a and step 3.b for each column that you want to add to the drill-through report.

4 **Optional:** To display a column name other than the name of the column that you selected in step 3:

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 4.a and step 4.b for each column for which you want to change the display name in the drill-through report.

5 **Optional:** Apply an aggregation function to drill-through columns.

Use an aggregation function to perform consolidations on the column values that are returned in the drill-through report.

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 5.a and step 5.b for each column to which you want to apply an aggregate function.

6 **Optional:** Define the sort order of a drill-through column.

See "Defining Sort Order for Drill-through Report Columns" on page 292 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:** 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 293 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 remaining tasks in the **Report Contents** tab of the drill-through report editor, see the following topics:

- "Defining Sort Order for Drill-through Report Columns" on page 292
- "Defining Template SQL" on page 293
- "Specifying Row Governors for Relational Sources" on page 294
- "Specifying Drill-through Report Filters for Relational Sources" on page 294
- "Testing Reports for Relational Sources" on page 295 and "Example Testing Scenarios —When Caption and Key Bindings Differ" on page 296

Defining Sort Order for Drill-through Report Columns

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 specify the sort order, spreadsheet users view data in the order determined by the external data source. Microsoft SQL Server, for example, sorts members in ascending order by default.

Note: This procedure is optional.

➤ To define the sort order of a drill-through column:

1 From the **Report Contents** tab of the drill-through report editor, 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.

2 Choose the **Asc** (ascending order) or **Desc** (descending order) option.

3 Repeat step 1 and step 2 for each column to which you want to apply a sort order.

4 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.

Defining Template SQL

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.

In the Template SQL dialog box, you define a template for drill-through SQL, referred to as "template SQL," which 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 "$$" syntax as the variable delimiter. Text contained within the $$ 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.

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 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 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 errors are corrected, click **OK** to return to the drill-through report editor.
8. To complete the remaining tasks in the **Report Contents** tab of the drill-through report editor, see the following topics:
   - "Defining Sort Order for Drill-through Report Columns" on page 292
   - "Specifying Row Governors for Relational Sources" on page 294
   - "Specifying Drill-through Report Filters for Relational Sources" on page 294
   - "Testing Reports for Relational Sources" on page 295 and "Example Testing Scenarios —When Caption and Key Bindings Differ" on page 296

Specifying Row Governors for Relational Sources

You specify row governors in the **Report Contents** tab of the **Drill-through Report Properties** dialog box.

Note: This procedure is optional.

➤ To specify a row governor, in the **Row governor** text box, enter the maximum number of rows to retrieve.

For example, to stop a query that retrieves more than 200 rows, enter `200`.

To complete the remaining tasks in the **Report Contents** tab of the drill-through report editor, see the following topics:

- "Defining Sort Order for Drill-through Report Columns" on page 292
- "Defining Template SQL" on page 293
- "Specifying Drill-through Report Filters for Relational Sources" on page 294
- "Testing Reports for Relational Sources" on page 295 and "Example Testing Scenarios—When Caption and Key Bindings Differ" on page 296

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:

   ```
   'contains'( connection : \'tbcSource'::'tbc.family'.'FAMILY'=="400")
   ```

   To filter a drill-through report for members in the "400" product family in the state of New York only, you may enter a filter such as the following;

   ```
   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 remaining tasks in the **Report Contents** tab of the drill-through report editor, see the following topics:
   - "Defining Sort Order for Drill-through Report Columns" on page 292
   - "Defining Template SQL" on page 293

- "Specifying Row Governors for Relational Sources" on page 294
- "Delayed Option Example" on page 297 and "Example Testing Scenarios—When Caption and Key Bindings Differ" on page 296

Testing Reports for Relational Sources

Use this procedure for testing drill-through reports based on relational sources. This procedure outlines a general method for testing. For other testing scenarios, see "Example Testing Scenarios —When Caption and Key Bindings Differ" on page 296 and "Example Testing Scenario— Recursive Hierarchies" on page 298.

➤ To test the drill-through report:

1. **In the Report Contents tab, click Test.**
2. **Select a Cube Schema 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. Under **Column Value**, enter a valid value for each column you select.

   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.

   When testing a recursive hierarchy, the **Intersection** and **Column Value** fields of the **Drill-through Report Testing** dialog box are used differently than when testing a standard hierarchy:

   - In the **Intersection** field, the recursive hierarchy name is displayed. For example, if the recursive hierarchy is named "recur_hier," then `recur_hier` is displayed.
   - In the **Column Value** field, a valid parent or child column name is entered, not a data value.

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 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 Testing dialog box.**

   To continue defining the drill-through report, select the **Associations** tab and perform the tasks in "Associating Drill-through Reports with Essbase Models" on page 306.

Example testing scenarios are provided in:

- "Example Testing Scenarios—When Caption and Key Bindings Differ" on page 296—where intersections contain elements that use the Advanced and Delayed key binding options
- "Example Testing Scenario—Recursive Hierarchies" on page 298—where a drill-through report includes a recursive hierarchy.

Example Testing Scenarios—When Caption and Key Bindings Differ

When testing a drill-through report, the value you enter in the Column Value field of the Drill-through Report Testing dialog must be a value that represents the key binding expression of the dimension element in the intersection.

In general, the value of the key binding expression of a dimension element is the same as the value of the caption binding expression, which represents an outline member name. However, some hierarchies that are used to create drill-through report intersections may have been built using dimension elements where the caption binding differs from the key binding. This is the case when the Advanced or Delayed options are selected for the key binding of a dimension element.

The following example procedures contain scenarios for testing drill-through reports when intersections contain elements that use the Advanced and Delayed options. See these topics:

- "Advanced Option Example" on page 296
- "Delayed Option Example" on page 297

Note: The example procedures assume you have created a data source connection to the sample TBC database.

Advanced Option Example

➤ To test with a hierarchy that uses the Advanced option in a dimension element:

1. Create a dimension element called SKU_Advanced, basing the caption binding expression on SKU from the TBC database, Product table (TBC.PRODUCT); for example:

   ```
   connection : \'tbcSource'::'TBC.PRODUCT'.'SKU'
   ```

2. Set the **Key Binding** type to **Advanced.**
3. Base the **Key Binding** expression on the PRODUCTID column from the Product table (TBC.PRODUCT); for example:

   ```
   connection : \'tbcSource'::'tbc.product'.'PRODUCTID'
   ```

4. Create a "product_advanced" hierarchy with the FAMILY column from the FAMILY table as the parent level, and the SKU_Advanced dimension element as the child.
5. Create a cube schema and Essbase model using the product_advanced hierarchy you created in step 4; use AMOUNT from TBC.SALES as the measure.
6. Create a drill-through report (**File**, then **New**, and then **Drill-through Report**) as specified below:

a. In the **Context** tab, add the product_advanced hierarchy.

b. In the **Report Contents** tab, place any fields from TBC.PRODUCT, for example: CAFFEINATED or PKGTYPE.

7 **Click Test to launch the Drill-through Report Testing dialog box.**

8 **Click the drop-down arrow in the first row under Intersection and select SKU_Advanced.**

9 **In Column Value, enter a value that represents the value shown in the key binding expression in step 3.**

Using the hierarchy created in this scenario, enter a value from the PRODUCTID column; for example:

```
1
```

10 **Click Show Result.**

Delayed Option Example

➤ To test with a hierarchy that uses the Delayed option in a dimension element:

1 **Create a dimension element called SKU_Delayed, basing the caption binding expression on SKU from the TBC database, Product table (TBC.PRODUCT); for example:**

```
connection : \'tbcSource'::'TBC.PRODUCT'.'SKU'
```

2 **Set the Key Binding type to Delayed.**

3 **Create a "product_delayed" hierarchy with the FAMILY column from the FAMILY table as the parent level, and the SKU_Delayed dimension element as the child.**

In the hierarchy, the key binding expression for the SKU_Delayed element should display as:

```
class : \'tbcSource'\'family'\'FAMILY'.'caption' || "_" || class :
\'DimElements'\'SKU_Delayed'.'caption'
```

4 **Create a cube schema and Essbase model using the "product_delayed" hierarchy you created in step 3; use AMOUNT from 'TBC.SALES' as the measure.**

5 **Create a drill-through report as specified below:**

a. In the **Context** tab, add the product_delayed hierarchy you created in step 3.

b. In the **Report Contents** tab, place any fields from TBC.PRODUCT, for example: CAFFEINATED or PKGTYPE.

6 **Click Test to launch the Drill-through Report Testing dialog box.**

7 **Click the drop-down arrow in the first row under Intersection and select SKU_Delayed.**

8 **In Column Value, enter values that represent the values that are shown in the key binding expression in step 3.**

Using the hierarchy created in this scenario, enter a value from the FAMILY column and a value from the SKU column, separated by an underscore. For example:

```
100_100-10
```

9 **Click Show Result.**

Example Testing Scenario—Recursive Hierarchies

When testing a recursive hierarchy, the Intersection and Column Value fields of the Drill-through Report Testing dialog box are used differently than when testing a standard hierarchy:

- In the **Intersection** field, the recursive hierarchy name is displayed.
- In the **Column Value** field, a parent or child column name is entered, not a data value.

This is illustrated in the following example scenario.

Note: The example procedure assumes (1) that you have created a data source connection to the sample TBC database, and (2) that you have created a minischema of the TBC source which contains a self-join between the PARENT and CHILD columns of the MEASURES table.

➤ To test a recursive hierarchy selected for drill-through:

1. **Create a recursive hierarchy called "recursive_hier" with the PARENT column from the MEASURES table as the parent level, and the CHILD column from the MEASURES table as the child level.**

   Note: Ensure that you have created a minischema which contains a self-join between the PARENT and CHILD columns of the MEASURES table

2. **Create a cube schema and Essbase model using the "recursive_hier" hierarchy you created in step 1; use AMOUNT from 'TBC.SALES' as the measure.**
3. **Create a drill-through report as specified below:**
   a. In the **Context** tab, add the "recursive_hier" hierarchy you created in step 1.
   b. In the **Report Contents** tab, place any fields from TBC.PRODUCT, for example: CAFFEINATED or PKGTYPE.
4. **Click Test to launch the Drill-through Report Testing dialog box.**
5. **Note that the Intersection field contains the name of the recursive hierarchy, "recursive_hier."**
6. **In the Column Value field, select a column name; for example, select COGS.**

   Note: A column name is used here, not column data.

7. **Click Show Result.**

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 FDM and Oracle BI EE 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: When you are using Spreadsheet Add-in, drill-through target URLs are limited to 256 characters in length. With Smart ViewSmart View, any limitations to URL length depend on the browser being used.

➤ 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, from Drill-through Report Type, select URL.**
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 300 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 301 for more information.

- **OBI URL Template**—Provides the syntax for a target Oracle BI EE URL based on the intersections you specified in the **Context** tab.

  See "Sample Oracle BI EE URL Template" on page 302 for more information.

3 Click **Save**, and 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 306.

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: When you are using Spreadsheet Add-in, drill-through target URLs are limited to 256 characters in length. With Smart ViewSmart View, any limitations to URL length depend on the browser being used.

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.

Syntax

The syntax assumes only two dimensions in the cube, Product and Market; and that the Product and Market hierarchies are selected in the Context tab of the drill-through report editor.

```
http://<server-name>/HyperionFDM/AuthorizedPages/IntersectionSummaryByLocation.aspx?
fdmAppName=<app-name>&fdmTargetAppName=<target-app-
name>&attribute=system.ds.essbase&attribute=server.ds.<server-ds>&attribute=app.ds.<app-
ds>&attribute=database.ds.<database.ds>&attribute=alias.ds.<alias-ds>&sso_token=$$CSS-
TOKEN$$&attribute=Product.id.$$Product-VALUE$$&attribute=Market.id.$$Market-
VALUE&rcp_version=1.5.0
```

Note: When you are using Spreadsheet Add-in, drill-through target URLs are limited to 256 characters in length. With Smart ViewSmart View, any limitations to URL length depend on the browser being used.

Parameters

- <server-name>—The name or IP address of the server and the port to which you want to connect.
- <app-name>—The FDM target 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.
- <target-app-name>—The name of the Essbase application from which the drill-through request is originating.
- <server-ds>—The name or IP address of the Essbase Server that is hosting the Essbase application and database from which the drill-through request to FDM is originating.
- <app-ds>—The name of the Essbase application from which the drill-through request is originating.
- <database.ds>—The name of the Essbase database from which the drill-through request is originating.
- <alias-ds>—The name of an alias table to use in the drill-through request. Use Default if no alias table is required.
- Product.id.$$Product-VALUE$$—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 as a part of the variable that is associated with the given dimension. Upon execution of the drill-through report, the name of the variable, $$Product-VALUE$$ will be substituted by the actual value from the POV; for example 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 Product in the Essbase Studio-generated syntax, as follows:

```
http://<server-name>/HyperionFDM/AuthorizedPages/IntersectionSummaryByLocation.aspx?
fdmAppName=<app-name>&fdmTargetAppName=<target-app-
name>&attribute=system.ds.essbase&attribute=server.ds.<server-
ds>&attribute=app.ds.<app-
ds>&attribute=database.ds.<database.ds>&attribute=alias.ds.<alias-ds>&sso_token=$
$CSS-TOKEN$$&attribute=Items.id.$$Product-VALUE$$&attribute=Market.id.$$Market-
VALUE&rcp_version=1.5.0
```

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.

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 URL example FDM URL drills through to a target FDM report containing the dimensions Scenario, Year, Accounts, Market, and Product.

The Product and Market 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://myfdmsvr:19000/HyperionFDM/AuthorizedPages/IntersectionSummaryByLocation.aspx?
fdmAppName=FDMAPP&fdmTargetAppName=Sample&attribute=system.ds.essbase&attribute=server.d
s.myfdmsvr11&attribute=app.ds.Sample&attribute=database.ds.Basic&attribute=alias.ds.Defa
ult&sso_token=$$CSS-TOKEN$$&attribute=Product.id.$$Product-VALUE$$&attribute=Market.id.$
$Market-VALUE$$&attribute=Year.id.$$Year-VALUE$$&attribute=Scenario.id.$$Scenario-VALUE$
$&attribute=Accounts.id.$$Accounts-VALUE$$&rcp_version=1.5.0
```

Sample Oracle BI EE URL Template

The Sample Oracle BI EE URL Template provides the syntax for a target Oracle BI EE 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?PortalPages&PortalPath=<portal-
path>&Action=Navigate&P0=2&P1=eq&P2=<Product-column>&P3=$$Product-VALUE$
$&P4=eq&P5=<Market-column>&P6=$$Market-VALUE$$
```

Note: When you are using Spreadsheet Add-in, drill-through target URLs are limited to 256 characters in length. With Smart ViewSmart View, any limitations to URL length depend on the browser being used.

Parameters

- `<server-name>`—The name or IP address of the server to which you want to connect.
- `<portal-path>`—Path to Oracle BI EE Presentation Service Report/Dashboard.
- `<number-of-parameters>`—Number of report parameters.
- `<Market-column>` and `<Product-column>`—The column in the Oracle BI EE 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.

  ```
  obi:http://obiserver:9704/analytics/saw.dll?
  PortalPages&PortalPath=paint&Action=Navigate&P0=2&P1=eq&P2=SKU&P3=$$Product-VALUE$
  $&P4=eq&P5=Region&P6=$$Market-VALUE$$
  ```

  Multiple columns may be specified.

Example

After selecting the Product and Market hierarchies in the Context tab of the drill-through report editor, upon executing the drill-through request, the "obi" prefix is removed from the target URL string and the following substitutions are made:

```
http://obiserver:9704/analytics/saw.dll?
PortalPages&PortalPath=paint&Action=Navigate&P0=2&P1=eq&P2=SKU&P3=100-
10&P4=eq&P5=Region&P6=East
```

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:

```
/**
 * 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.

```
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);
      result.addTag("Class", String.class);
      result.addTag("Value", String.class);

      Iterator<String[]> iterator = args.iterator();

      while (iterator.hasNext())
      {
         // add records to the result

         String[] tuple = iterator.next();

         result.setValue("Hierarchy", tuple[0]);
         result.setValue("Class", tuple[1]);
         result.setValue("Value", tuple[2]);

         result.addRecord();
      }
   }
}
```

When the spreadsheet user runs the drill-through report, the Java method is executed and results in the following drill-through report:

```
======================================
Hierarchy    Class       Value
======================================
ProductH     SKU         100-10
TimeH        Time Year   2006
```

```
MarketH      REGION       East
MyMeasuresH  Msr_Gen4     Sales
=======================================
```

4 Click **Save**, and 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 306.

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 editor, select 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 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 364 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 **Refresh model list.**

3 Click **Save**; then **Close** to close the drill-through report editor.

17

Find and Search

In This Chapter

About Find and Search

Essbase Studio offers two methods to help you locate metadata elements, Find and Search.

Find focuses on looking for metadata within selected user interface components of the Essbase Studio Console. For example, you may choose to find a metadata element in the **Metadata Navigator**, the minischema work area, or the **Data Source Navigator**. Further, you may select a folder in the **Metadata Navigator** or **Data Source 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.

See "Finding Metadata Elements" on page 307 and "Searching for Metadata Elements" on page 308.

Finding Metadata Elements

Use Find to locate a metadata element within a UI component, such as the **Metadata Navigator**, the Minischema pane, or the **Data Source 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, in the **Data Source Navigator**, expand a data source connection, expand the **Minischemas** folder, and then right-click a minischema.and select **Edit**.

2 **Select Edit, then 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 a minischema containing the string, `product`, type product in the **Find** text box.

4 **Optional: Select options to refine the Find:**

- **Match Case**—Find only metadata elements that exactly match your find string.

  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 elements that 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 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 metadata elements library is large.

➤ To search for metadata elements:

1 **Select Edit, then 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, using the drop-down list, 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**. 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 box.

   The entire contents 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.

18 Preferences

In This Chapter

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**, then **Preferences.**

2 To display a confirmation message when deleting items from the **Data 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 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**, then **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**—Enables join lines to be drawn diagonally between tables, where appropriate.
- **Rectangle**—Enables 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**.

Setting Essbase Model Preferences

Set Essbase model preferences to control certain user prompts.

➤ To set Essbase model preferences:

1 Select **Edit**, then **Preferences.**

2 Select the following check boxes to enable the specified action:

- **Ask user if the model should be tuned when setting aggregate storage to true**
- **Ask user to validate a changed model before closing the Essbase Model Properties dialog box**

Clear the check box to disable a particular prompt.

3 Click **Apply** to apply your selections, then click **OK.**

Note: To return to the default preference settings, click **Restore Defaults**.

19 Integration Services Catalog Migration

In This Chapter

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 . (In Integration Services, this is referred to as the OLAP Metadata 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 313
- "Metaoutline Migration" on page 314

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 a 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 314
- “Mapping the Measure Column” on page 314
- “Creating the Essbase Export Model” on page 315
- “Mapping Aliases, User-defined Members, and Reports” on page 315

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 315.
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.**

View the migration progress in the **Progress Information** dialog box.

12 **Optional: To cancel the migration:**

a. Press **Cancel**.

b. Delete any objects that are present in the target directory you specified in step 10.

Your OLAP 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 **Data Source Navigator**.

Limitations and Restrictions

The Essbase Studio migration utility has the following limitations and restrictions.

Guidelines for Migrating 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 with multiple ODBC DSNs are not migrated.
- Metaoutlines with drill-through defined on recursive hierarchies are supported. After migration, those recursive hierarchy drill-through definitions can be used in Essbase Studio.
- 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.
- If a data source column is bound to a level-0 member set, and contains both base members and shared members, it is possible that the data aggregated in the Essbase Studio-generated data load SQL for the specific base members (which have shared members in the same dimension) will be augmented at times, depending on the number of shared members. In this case, Oracle recommends that users define their own custom data load SQL as described in "Overriding Standard Data Load SQL" on page 187. This behavior is true both for Essbase models created from metaoutlines imported from Integration Services, and for Essbase models created from the beginning in Essbase Studio.

  For example, using the TBC sample, the data aggregated for 100-10 in the standard data load SQL is twice as much as it should be (doubled) because 100-10 has shared members under "Diet". To get the correct aggregated data, users can select the **Use Custom SQL for data load** option, and manually add one more "group by" column, "Family", to the custom data load SQL.
- There are differences in attribute member handling between Integration Services and Essbase Studio.

  Integration Server adds all attribute members of an attribute dimension, whether or not the attribute association relationship exists in the corresponding base dimension. To do this,Integration Server produces three types of rules files during member load: one for the base dimension, one for the attribute dimension, and one for the attribute associations.

  Essbase Studio creates attribute members only when the corresponding attribute association relationships exist in the base dimension. During cube deployment, it produces just two rules files: one for the base dimension, and one for attribute members and their associations. This helps to speed up cube deployment, but those attribute members without a base member association are dropped in the process.

  For example, an outline created by Integration Services using the TBC sample metaoutline has the attribute dimension "Population" with these members:

  ```
  Small: 3000000, 6000000
  Medium: 900000, 1200000, 1500000, 1800000
  Large: 21000000, 24000000, 27000000, 30000000, 33000000
  ```

  However, if the TBC metaoutline is migrated to the Essbase Studio catalog and deployed to Essbase Server, the new outline will have the attribute dimension "Population" with these members:

  ```
  Small: 3000000, 6000000
  Medium: 900000, 1200000, 1500000
  Large: 21000000, 33000000
  ```

  This is because there are no states with the population values 18000000, 24000000, 27000000, or 30000000.

Note that if the attribute dimension is built in a recursive way (from a parent-child table), Essbase Studio will produce three rules files—a for the base dimension, the attribute dimension, and an attribute association rules file—just as is done in Integration Services. As a result, the attribute dimension will have all members regardless of association.

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 data load 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.

A

Accessibility

In This Appendix

About Essbase Studio Accessibility

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 for the keyboard equivalents to work.

Keyboard equivalents common to Microsoft Windows, such as Ctrl+S for Save, are not documented.

Note: For users requiring high contrast, Essbase Studio works best with these schemes: Windows Classic, Windows Standard, and High Contrast White.

Note: If you are using JAWS® Screen Reading Software, Oracle recommends using the Microsoft Internet Explorer browser.

See these sections for information on the various keyboard equivalents available in Essbase Studio:

- "Keyboard Equivalents for Access and Navigation" on page 321
- "Keyboard Equivalents in Dialog Boxes and Editors" on page 326
- "Keyboard Equivalents in the Minischema Editor" on page 350

Keyboard Equivalents for Access and Navigation

The tables in this section describe keyboard equivalents that assist with accessing Essbase Studio Console elements and navigating through the interface:

Table 6 Accessing Main Areas of the Essbase Studio Interface

Action	Key or Key Combination
Access to the Essbase Studio menu bar.	**F10** **Note:** Press **Esc** to remove focus from the menu bar.
Focus on **Metadata Navigator**.	**Ctrl**+**5** Or: **Alt**+**V**, then **N** Or: Press and hold down **Ctrl**, press **F7**, use up and down arrow keys to select **Metadata Navigator**, and then press **Enter**
Focus on **Data Source Navigator**.	**Ctrl**+**3** Or: **Alt**+**V**, then **A** Or: Press and hold down **Ctrl**, press **F7**, use up and down arrow keys to select **Data Sources**, and then press **Enter**
Focus on **Console Messages** tab.	**Ctrl**+**2** Or: **Alt**+**V**, then **C** Or: Press and hold down **Ctrl**, press **F7**, use up and down arrow keys to select **Console Messages**, and then press **Enter**
Display the context menu appropriate to the focus of the cursor.	**Shift**+**F10**
Close, without saving, an open dialog box or wizard.	**Esc**

Table 7 Accessing Elements in the **Metadata Navigator** and **Data Source Navigator**

Action	Key or Key Combination
Navigate through tree structure.	**Ctrl**+**5** or **Ctrl**+**3**, then use up and down arrow keys
Open a folder.	Right arrow key
Expand all folders in the **Metadata Navigator**.	Use the up and down arrow keys to navigate to the **Root** folder, **Shift**+**F10**, then **E**
Close a folder.	Left arrow key
Collapse all folders in the **Metadata Navigator**.	Use the up and down arrow keys to navigate to the **Root** folder, **Shift**+**F10**, then **C**
Refresh the contents of a folder.	Use the up and down arrow keys to navigate to a folder, **Shift**+**F10**, then **H**
Refresh all folders in the **Metadata Navigator**.	Use the up and down arrow keys to navigate to the **Root** folder, **Shift**+**F10**, then **H**
Context menu for tree items.	**Shift**+**F10**
Launch the wizard, editor, or properties dialog box for the selected metadata element.	Use arrow keys to navigate to element location in tree, press **Enter**
Open the **Search** tab in the **Metadata Navigator**.	**Alt**+**E**, then **S**
Open the **Find** dialog box.	**Alt**+**E**, then **F**
Launch the editor or **Properties** dialog box for the selected source element in the **Data Source Navigator** (applies to data source connection names, Essbase connection names, and minischemas).	**Ctrl**+**3**, use arrow keys to navigate to the element, **Shift**+**F10**, then **E**
Launch the **Properties** dialog box for the selected source element (applies to source table names, column names, and minischemas).	**Ctrl**+**3**, use arrow keys to navigate to the element, **Shift**+**F10**, then **O**
Launch sample data for the selected metadata element in the **Metadata Navigator**.	**Ctrl**+**5**, use arrow keys to navigate to the metadata element, **Shift**+**F10**, then **V**
Launch sample data for the selected source column element in the **Data Source Navigator**.	**Ctrl**+**3**, use arrow keys to navigate to the column element , **Shift**+**F10**, then **V**
Launch the lineage editor in the context of the selected metadata element in the **Metadata Navigator**.	**Ctrl**+**5**, use arrow keys to navigate to the element, **Shift**+**F10**, then **L**
Launch the graphical view of the Essbase model in the work area.	**Ctrl**+**5**, use arrow keys to navigate in the **Metadata Navigator**, select an Essbase model, then press **Enter**
Launch the **Essbase Model Properties** dialog box.	**Ctrl**+**5**, use arrow keys to navigate in the **Metadata Navigator**, select an Essbase model, **Shift**+**F10**, then **B**

Action	Key or Key Combination
Launch the **Cube Deployment Wizard.**	**Ctrl**+**5**, use arrow keys to navigate in the **Metadata Navigator**, select an Essbase model, then: **Shift**+**F10**, then **W** Or **Alt**+**F**, then **W**
Launch the deployment history in the work area.	**Ctrl**+**5**, use arrow keys to navigate to a element in the **Metadata Navigator**, select an Essbase model, **Shift**+**F10**, then **Y** for **Show Deployment History**
Launch the **Incremental Update** dialog box.	**Ctrl**+**3**, use arrow keys to navigate to a data source connection, **Shift**+**F10**, then **U**
Launch the **Introspection Wizard**.	**Ctrl**+**3**, use arrow keys to navigate to a data source connection, **Shift**+**F10**, then **I**
Microsoft SQL Server users only: Toggle the **Show Friendly Names** option	**Ctrl**+**3**, use arrow keys to navigate to a Microsoft SQL Server data source connection, **Shift**+**F10**, then **Y**

Table 8 Right-Click Commands Common to Most Elements in the Metadata Navigator

Action	Key or Key Combination
Edit an element.	**Shift**+**F10**, then **E**
Copy an element.	**Shift**+**F10**, then **C**
Paste an element.	**Shift**+**F10**, then **P**
Rename an element.	**Shift**+**F10**, then **N**
Delete an element.	**Shift**+**F10**, then **D**
Refresh cube schema or Essbase model	**Shift**+**F10**, then **H**

Table 9 Accessing Editors, Wizards, and Dialog Boxes to Create New Elements

Action	Key or Key Combination
Launch **Connection Wizard**.	**Alt**+**F**, followed by **N**, then **A** Or: **Alt**+**F**, followed by **N**, then **Enter** (the default selection is for **Connection Wizard**)
Launch **Minischema Wizard**.	**Alt**+**F**, followed by **N**, then **I**
Launch the **User-Defined Table** dialog box.	**Alt**+**F**, followed by **N**, then **U**
Launch the **Folder Properties** dialog box.	**Ctrl**+**5**, use arrow keys to navigate to appropriate location in the tree, **Alt**+**F**, followed by **N**, then **F**
Launch the **Dimension Element** dialog box.	**Ctrl**+**5**, use arrow keys to navigate to appropriate location in the tree, **Alt**+**F**, followed by **N**, then **D**

Action	Key or Key Combination
Launch the **Derived Text Measure** dialog box.	**Ctrl**+**5**, use arrow keys to navigate to appropriate location in the tree, **Alt**+**F**, followed by **N**, then **T**
Launch the hierarchy editor.	**Ctrl**+**5**, use arrow keys to navigate to appropriate location in the tree, **Alt**+**F**, followed by **N**, then **H**
Launch the measure hierarchy editor.	**Ctrl**+**5**, use arrow keys to navigate to appropriate location in the tree, **Alt**+**F**, followed by **N**, then **M**
Launch the **Calendar Hierarchy** dialog box.	**Ctrl**+**5**, use arrow keys to navigate to appropriate location in the tree, **Alt**+**F**, followed by N, then C
Launch **Cube Schema Wizard**.	**Ctrl**+**5**, use arrow keys to navigate to appropriate location in the tree, **Alt**+**F**, followed by **N**, then **S**
Launch the drill-through report editor.	**Ctrl**+**5**, use arrow keys to navigate to appropriate location in the tree, **Alt**+**F**, followed by **N**, then **R**
Launch the **Create Date Elements** dialog box.	**Ctrl**+**5**, use arrow keys to navigate in the tree to select a date column, **Shift**+**F10**, then **T**

Table 10 Accessing Dialog Boxes from the Menu Bar

Action	Key or Key Combination
Launch the **Export** dialog box.	**Alt**+**F**, then **O**
Launch the **Import** dialog box	**Alt**+**F**, then **I**
Open the **Find** dialog box.	**Alt**+**E**, then **F**
Open the **Search** tab in the **Metadata Navigator**.	**Alt**+**E**, then **S**
Open the **Properties** dialog box for the select metadata or source element.	**Alt**+**E**, then **R**
Launch the **Preferences** dialog box.	**Alt**+**E**, then **E**
Launch the **Essbase Studio Login** dialog box to reconnect to Essbase Studio Server.	**Alt**+**T**, then **R**
Launch the **Rehost Essbase Connections** dialog box.	**Alt**+**T**, then **H**
Launch the **Alias Set Manager**.	**Alt**+**T**, then **A**
Launch the **Update Cube Linkage** dialog box.	**Alt**+**T**, then **U**
Launch the **EIS Catalog Migration** dialog box.	**Alt**+**T**, then **E**

Table 11 Working with Editors in the Work Area

Action	Key or Key Combination
Move from editor to editor	With focus on editor title tab text, press and hold **Ctrl**, then **F6**, use up and down arrow keys to choose editor from the context menu. You must hold the **Ctrl** key down for the entire operation.
Invoke context menu for editor	**Alt**+**-**

Action	Key or Key Combination
Within drill-through report editor, move from tab to tab	Right or left arrow keys
Close an editor	**Ctrl**+**F4** Or **Alt**+**F**, then **C** Or **Ctrl**+**W**
Close multiple editors	**Alt**+**F**, then **L** Or **Ctrl**+**Shift**+**W**

Table 12 Accessing **Help** Menu Items

Action	Key or Key Combination
Launch the **Welcome** page in the work area	**Alt**+**H**, then **W**
Launch the Essbase Studio online help	**Alt**+**H**, then **H** Or Within dialog boxes, editors, wizards, and main console panes, press **F1**. **Note:** To close the online help, press **Alt**+**F4**.
Launch the Oracle Technical Support Web site	**Alt**+**H**, then **T**
Launch the EPMOracle Hyperion Enterprise Performance Management System Documentation Library on the Oracle Technology Network Web site	**Alt**+**H**, then **E**
Launch the Essbase Studio About box	**Alt**+**H**, then **A**

Keyboard Equivalents in Dialog Boxes and Editors

The tables in this section describe keyboard equivalents found in Essbase Studio Console dialog boxes and editors.

- Table 13, "Keyboard Equivalents for the **Essbase Studio Login** Dialog Box," on page 329
- Table 14, "Keyboard Equivalents for **Connection Wizard**, **Define Connection** Page," on page 330
- Table 15, "Keyboard Equivalents for **Connection Wizard**, **Select Tables** Page," on page 331
- Table 16, "Keyboard Equivalents for **Connection Wizard**, **Model Files** Page (text file data sources only)," on page 331

Table 13 Keyboard Equivalents for the **Essbase Studio Login** Dialog Box

Element Name	Equivalent
Server	Alt+S
User	Alt+U

Element Name	Equivalent
Password	**Alt+P**
Remember this user	**Alt+R**

Note: The **Essbase Studio Login** dialog box is also launched when users select **Tools**, then **Reconnect To Server** (Alt+T, then R).

Table 14 Keyboard Equivalents for **Connection Wizard**, **Define Connection** Page

Element Name	Equivalent
Connection name	**Alt+C**
Connection Description	**Alt+O**
Data Source Type	**Alt+D**
Presentation Layer (Oracle BI EE sources only)	**Alt+L**
Business Model (Oracle BI EE sources only)	**Alt+M**
Essbase Server (Essbase Server connections only)	**Alt+V**
Server Name	**Alt+V**
Default (port number)	**Alt+E**
Cluster (Essbase Server connections only)	**Alt+L**
SID	**Alt+I**
Service Name	**Alt+R**
Pool Max Size	**Alt+X**
Cache Size	**Alt+Z**
Authentication Method	**Alt+H**
User Name	**Alt+U**
Password	**Alt+P**
Database Name	**Alt+A**
Browse (for text file location)	**Alt+W**
Skip records (text file sources only)	**Alt+K**
Column names in first row (text file sources only)	**Alt+R**
Comma (text file sources only)	**Alt+M**

Element Name	Equivalent
Tab (text file sources only)	**Alt**+**A**
Space (text file sources only)	**Alt**+**P**
Custom (text file sources only)	**Alt**+**U**
Test Connection	**Alt**+**T**

Table 15 Keyboard Equivalents for **Connection Wizard**, **Select Tables** Page

Element Name	Equivalent
Filter	**Alt**+**T**
Apply	**Alt**+**A**
Lock catalog during exploration	**Alt**+**L**

Table 16 Keyboard Equivalents for **Connection Wizard**, **Model Files** Page (text file data sources only)

Element Name	Equivalent
Select all	**Alt**+**S**
Clear all	**Alt**+**C**
Filter	**Alt**+**T**
Skip records	**Alt**+**K**
Column names in first row	**Alt**+**R**
Comma	**Alt**+**M**
Tab	**Alt**+**A**
Space	**Alt**+**P**
Custom	**Alt**+**U**
Edit Column Properties	**Alt**+**E**

Table 17 Keyboard Equivalents for **Column Properties** Dialog Box (accessed from the **Connection Wizard**, **Model Files** Page)

Element Name	Equivalent
Name	**Alt**+**M**
Data type	**Alt**+**D**, then use the arrow keys to select an option: ● **Text** ● **Integer** ● **Large integer** ● **Decimal**

Element Name	Equivalent
Back	**Alt+B**
Next	**Alt+N**

Table 18 Keyboard Equivalents for **Connection Wizard**, **Select Minischema** Page

Element Name	Equivalent
Create a new schema diagram	**Alt+C**
Clear existing schema	**Alt+X**
Skip minischema diagram	**Alt+S**
Use Introspection to Detect Hierarchies	**Alt+I**

Table 19 Keyboard Equivalents for **Connection Wizard**, **Populate Minischema** Page

Element Name	Equivalent
Connections	**Alt+C**
Databases	**Alt+D**
Filter	**Alt+T**
Apply	**Alt+A**
Add Related Objects	**Alt+R**

Table 20 Keyboard Equivalents for **Connection Wizard**, **Create Metadata Elements** Page or **Create OBIEE Dimension Elements** Page

Element Name	Equivalent
Browse	**Alt+W**

Table 21 Keyboard Equivalents for **Minischema Wizard**, **Name and Description** Page

Element Name	Equivalent
Minischema Name	**Alt+M**
Minischema Description	**Alt+D**

Table 22 Keyboard Equivalents for **Minischema Wizard**, **Add/Remove Tables** Page

Element Name	Equivalent
Connections	**Alt+C**
Filter	**Alt+T**
Apply	**Alt+A**

Element Name	Equivalent
Add Related Tables	**Alt+R**

Table 23 Keyboard Equivalents for **Edit Properties of Minischema Join** Dialog Box

Element Name	Equivalent
Table	**Alt+T**
Column	**Alt+C**
Outer	**Alt+O**
Left	**Alt+L**
Full	**Alt+F**
Right	**Alt+R**

Table 24 Keyboard Equivalents for **Create Joins by Inspection** Dialog Box

Element Name	Equivalent
Table name filter	**Alt+T**
Column name filter	**Alt+C**
Select all items	**Alt+S**

Table 25 Keyboard Equivalents for **User-Defined Table Definition** Dialog Box

Element Name	Equivalent
Connection	**Alt+C**
Table name	**Alt+T**
Table definition	**Alt+D**

Table 26 Keyboard Equivalents for **Rehost Essbase Connections** Dialog Box

Element Name	Equivalent
Update the host name and port number for all deployment history	**Alt+D**
Replicate the last successful deployment history and update the copy only	**Alt+R**
Test Connection	**Alt+T**
Update	**Alt+U**
Close	**Alt+C**

Table 27 Keyboard Equivalents for **Preferences** Dialog Box, **General**

Element Name	Equivalent
Show Confirm Delete dialog	**Alt+C**
Show informational messages	**Alt+I**
Show warning messages	**Alt+W**
Show error messages	**Alt+E**
Browse	**Alt+B**
Restore Defaults	**Alt+D**
Apply	**Alt+A**

Table 28 Keyboard Equivalents for **Preferences** Dialog Box, **Schema**

Element Name	Equivalent
Column type indicator	**Alt+T**
Primary Key	**Alt+P**
Foreign Key	**Alt+F**
Diagonal	**Alt+G**
Rectangle	**Alt+R**
Arrow	**Alt+O**
Crow's Feet	**Alt+W**
Restore Defaults	**Alt+D**
Apply	**Alt+A**

Table 29 Keyboard Equivalents for **Preferences** Dialog Box, **Essbase Model**

Element Name	Equivalent
Ask user if the model should be tuned when setting aggregate storage to true	**Alt+M**
Ask user to validate a changed model before closing the Properties dialog box	**Alt+V**

Table 30 Keyboard Equivalents for **Folder Properties** Dialog Box

Element Name	Equivalent
Name	Alt+N
Description	Alt+D

Table 31 Keyboard Equivalents for **Dimension Element Properties** Dialog Box

Element Name	Equivalent
Name	**Alt+N**
Description	**Alt+D**
Formula	**Alt+F**
Caption Binding	**Alt+C**
Key Binding option: **Same as caption**	**Alt+S**
Key Binding option: **Advanced**	**Alt+A**
Key Binding option: **Delayed**	**Alt+Y**
Filter	**Alt+T**
Sort Order	**Alt+O**

Table 32 Keyboard Equivalents for **Derived Text Measure Properties** Dialog Box

Element Name	Equivalent
Name	**Alt+N**
Description	**Alt+D**
Formula	**Alt+F**
Expression	**Alt+E**
Range	**Alt+R** **Note:** Use **Alt+R** when rows are present in the **Range** table. When the **Range** table is empty, tab to the **Add another row** button to begin adding rows to the table.

Table 33 Keyboard Equivalents for **Text List Properties** Dialog Box

Element Name	Equivalent
Name	**Alt+N**
Description	**Alt+D**
Bindings	**Alt+B**
Value Binding	**Alt+V**
ID Binding	**Alt+I**

Table 34 Keyboard Equivalents for Hierarchy and Measure Hierarchy Editor

Element Name	Equivalent
Name	**Alt**+**N**
Description	**Alt**+**D**
Data	**Alt**+**T**
Add	**Alt**+**A**
Preview With Caption Binding (not available for measure hierarchies)	**Alt**+**P**+**C**
Preview With Key Binding (not available for measure hierarchies)	**Alt**+**P**+**K**
Add child	**Alt**+**A** Or **Insert**
Add sibling	**Alt**+**S** Or **Shift**+**Insert**
Add user-defined child	**Alt**+C Or **Ctrl**+**Insert**
Add user-defined sibling	**Alt**+**U** Or **Ctrl**+**Shift**+**Insert**

Table 35 Keyboard Equivalents for **Calendar Hierarchy** Dialog Box

Element Name	Equivalent
Hierarchy Name	**Alt**+**H**
Gregorian	**Alt**+**G**
Fiscal	**Alt**+**F**
Retail	**Alt**+**R**
ISO	**Alt**+**I**
Manufacturing	**Alt**+**M**
Start Date	**Alt**+**S**
End Date	**Alt**+**D**
First day of week	**Alt**+**W**

Element Name	Equivalent
Time Depth	**Alt+T**
Edit	**Alt+E**
Day Attributes	**Alt+A**
Edit LVA	**Alt+V**

Table 36 Keyboard Equivalents for **Day Attributes** Dialog Box

Element Name	Equivalent
Sunday	**Alt+S**
Monday	**Alt+M**
Tuesday	**Alt+T**
Wednesday	**Alt+W**
Thursday	**Alt+H**
Friday	**Alt+F**
Saturday	**Alt+U**
Dimension Name	**Alt+D**
Add	**Alt+A**
Remove	**Alt+R**

Table 37 Keyboard Equivalents for **Edit Linked Value Attributes** Dialog Box

Element Name	Equivalent
Linked Value Attributes	**Alt+L**, then use the **Tab** key to tab through the options.

Table 38 Keyboard Equivalents for **Edit Labeling Rules**

Element Name	Equivalent
Use 2-digit *time entity*	**Alt+U**

Table 39 Keyboard Equivalents for **Create Date Elements** Dialog Box

Element Name	Equivalent
Year	**Alt+Y**
Quarter	**Alt+Q**
Month	**Alt+M**

Element Name	Equivalent
Day of week	**Alt+D**

Table 40 Keyboard Equivalents for **Alias Set Manager**

Element Name	Equivalent
Alias sets	**Alt+A**
Description	**Alt+D**
Bindings	**Alt+B**
Bindings popup menu	**Alt+B**, **Shift+F10** The **Bindings** popup menu has these options: ● **N—Sort by dimension** element name ● **P—Sort by dimension element path** ● **B—Sort by bindings** ● **A—Sort ascending** ● **D—Sort descending** ● **F—Show full path for dimension elements**

Table 41 Keyboard Equivalents for the **Create Bindings** Dialog Box

Element Name	Equivalent
Source selection tree	**Alt+S**
Binding expression	**Alt+X**
Binding expression tree	**Alt+B**

Table 42 Keyboard Equivalents for **Cube Schema Wizard, Choose Measures and Hierarchies** Page

Element Name	Equivalent
Cube Schema Name	**Alt+S**
Comment	**Alt+C**
Available Dimension Elements	**Alt+A**
Measures/Measures Hierarchy	**Alt+M**
Hierarchies	**Alt+H**
Preview Hierarchies	**Alt+P**

Table 43 Keyboard Equivalents for **Cube Schema Wizard**, **Cube Schema Options** Page

Element Name	Equivalent
Override default data load bindings	**Alt+O**
Create Essbase Model	**Alt+C**
Model Name	**Alt+M**
Account Dimension	**Alt+D**
Model Type: Aggregate Storage	**Alt+A**
Model Type: Block Storage	**Alt+S**

Table 44 Keyboard Equivalents for **Default Load Binding** Dialog Box

Element Name	Equivalent
Formula	**Alt+U**
Expression	**Alt+E**
Filter	**Alt+F**

Note: To access the **Default Load Binding** dialog box: In the **Define data load mappings** page of the **Cube Schema Wizard**, use the up and down arrow keys to navigate to the leaf level of a hierarchy, use the right arrow key to move to the **Data Load Binding** column, and press **Enter** to activate the cell. Press **Enter** again to activate the ellipsis button and launch the **Default Load Binding** dialog box.

Table 45 Keyboard Equivalents for **Essbase Model Properties**, **General** Tab

Element Name	Equivalent
Comment	**Alt+M**
Last deployed	**Alt+L**
Aggregate storage model	**Alt+A**
Duplicate member name support	**Alt+D**
Use Unicode character set	**Alt+U**
XOLAP Model	**Alt+X**
Custom data load settings	**Alt+S**, launches the **Define Data Load Settings** dialog box
Date format	**Alt-F**

Table 46 Keyboard Equivalents for **Essbase Model Properties**, **Alias** Tab

Element Name	Equivalent
Available Tables	Alt+A
Selected Tables	Alt+S

Table 47 Keyboard Equivalents for **Essbase Model Properties**, **Attributes** Tab

Element Name	Equivalent
Value	Alt+V
Prefix/Suffix	Alt+P
Separator	Alt+S
Dimension	Alt+D
Sum member	Alt+U
Count member	Alt+T
Minimum member	Alt+M
Maximum member	Alt+X
Average member	Alt+G
True members	Alt+E
Dates are	Alt+A
False members	Alt+F
Numerics are	Alt+R

Table 48 Keyboard Equivalents for **Essbase Model Dimension Properties**, **General** Tab

Element Name	Equivalent
Name in Cube	Alt+N
Comment	Alt+M
Edit Named Generations and Levels	Alt+E

Table 49 Keyboard Equivalents for **Essbase Model Dimension Properties**, **Edit Named Generational and Levels** Dialog Box

Element Name	Equivalent
Named Generations	Alt+G
Named Levels	Alt+L

Table 50 Keyboard Equivalents for **Essbase Model Dimension Properties**, **Info** Tab

Element Name	Equivalent
Dimension Type	Alt+T
Dynamic series	Alt+Y
Dimension Storage	Alt+D
Two Pass Calculation	Alt+P
Data Storage	Alt+S
Dimension Solve Order	Alt+V
Member Solve Order	Alt+M

Table 51 Keyboard Equivalents for **Essbase Model Dimension Properties**, **Account Info** Tab

Element Name	Equivalent
Time Balance	Alt+T
Skip	Alt+S
Variance Reporting	Alt+V

Table 52 Keyboard Equivalents for **Essbase Model Dimension Properties**, **Formula** Tab

Element Name	Equivalent
Formula	Alt+F

Table 53 Keyboard Equivalents for **Essbase Model Dimension Properties**, **UDAs** Tab

Element Name	Equivalent
UDA value	Alt+U
Add to list	Alt+A
Existing UDAs	Alt+E

Table 54 Keyboard Equivalents for **Essbase Model Dimension Properties**, **Outline Build** Tab

Element Name	Equivalent
Hierarchy settings	Alt+H
Move duplicate member settings	Alt+M
Optimize data load by using a fact table column as data source binding	Alt+Z
Reverse position of shared and actual members if shared member is located before actual member	Alt+R

Table 55 Keyboard Equivalents for **Essbase Model Member Properties**, **General** Tab

Element Name	Equivalent
Comments	**Alt+M**
Essbase attribute for <*member name*>	**Alt+A**
Attribute dimension name	**Alt+D**
Attribute type	**Alt+T**
Edit Varying Attribute Settings	**Alt+E**
Edit Named Generations and Levels	**Alt+I**

Table 56 Keyboard Equivalents for **Essbase Model Properties**, **Edit Varying Attributes** Dialog Box

Element Name	Equivalent
Create as Varying Attribute	**Alt+C**
Association mode	**Alt+A**
Independent Dimension Settings	**Alt+I**

Table 57 Keyboard Equivalents for **Essbase Model Member Properties**, **Info** Tab

Element Name	Equivalent
Consolidation	**Alt+D**
Solve Order	**Alt+V**
Aggregate Storage Options	**Alt+A**
Two Pass Calculation	**Alt+P**
Data Storage	**Alt+S**

Table 58 Keyboard Equivalents for **Essbase Model Member Properties**, **Account Info** Tab

Element Name	Equivalent
Time Balance	**Alt+T**
Variance Reporting	**Alt+V**
Skip	**Alt+S**

Table 59 Keyboard Equivalents for **Essbase Model Member Properties**, **Typed Measures** Tab

Element Name	Equivalent
Member format string	**Alt+F**

Element Name	Equivalent
Current Assigned Text List	**Alt+A**
Clear Current Assigned Text List	**Alt+L**

Table 60 Keyboard Equivalents for **Essbase Model Member Properties**, **Formula** Tab

Element Name	Equivalent
Formula	**Alt+F**

Table 61 Keyboard Equivalents for **Essbase Model Member Properties**, **Alias** Tab

Element Name	Equivalent
Rules for mapping text replacement	**Alt+R**
Create rule	**Alt+A**
Edit details	**Alt+E**
Delete	**Alt+D**
Change case to	**Alt+H**
Drop leading and trailing spaces	**Alt+L**
Convert spaces to underscores	**Alt+V**
Prefix	**Alt+P**
Prefix Separator	**Alt+F**
Custom prefix	**Alt+X**
Suffix	**Alt+S**
Suffix separator	**Alt+U**
Custom suffix	**Alt+M**

Table 62 Keyboard Equivalents for **Essbase Model Member Properties**, **Alias** Tab—**Search Rule** Dialog Box

Element Name	Equivalent
Search for	**Alt+S**
Replace with	**Alt+R**
Case sensitive	**Alt+C**
Match whole word	**Alt+M**
Replace all occurrences	**Alt+A**

Table 63 Keyboard Equivalents for **Essbase Model Member Properties**, **UDAs** Tab

Element Name	Equivalent
UDA value	Alt+U
External source	Alt+X
Existing UDAs	Alt+E
UDA Value—Add to list	Alt+A
External Source—Add to list	Alt+D

Table 64 Keyboard Equivalents for **Essbase Model Member Properties**, **Transformations** Tab

Element Name	Equivalent
Prefix	Alt+P
Prefix separator	Alt+F
Custom prefix	Alt+T
Suffix	Alt+S
Suffix separator	Alt+I
Custom suffix	Alt+X
Rules for text replacement	Alt+R
Create rule	Alt+U
Edit details	Alt+E
Delete	Alt+D
Drop leading and trailing spaces	Alt+A
Convert spaces to underscores	Alt+V

Table 65 Keyboard Equivalents for **Essbase Model Member Properties**, **Transformation** Tab—**Search Rule** Dialog Box

Element Name	Equivalent
Search for	Alt+S
Replace with	Alt+R
Case sensitive	Alt+C
Match whole word	Alt+M
Replace all occurrences	Alt+A

Table 66 Keyboard Equivalents for **Cube Deployment Wizard**, **Essbase Server Connection Options** Page

Element Name	Equivalent
Essbase Server Connection	Alt+E
Application	Alt+A
Database	Alt+D
New Connection	Alt+W
ODBC (Essbase dynamically creates ODBC connection string)	Alt+O
ODBC (Enter ODBC DSN name)	Alt+C
OCI (Enter OCI connection identifier)	Alt+I
Model Properties	Alt+M

Table 67 Keyboard Equivalents for **Essbase Login** Dialog Box (accessed when creating an Essbase Server connection in the **Cube Deployment Wizard**)

Element Name	Equivalent
Name	Alt+N
Description	Alt+D
Server	Alt+S
Port	Alt+T
User	Alt+U
Password	Alt+P
Cluster	Alt+C

Table 68 Keyboard Equivalents for **Cube Deployment Wizard**, **Cube Deployment Options** Page

Element Name	Equivalent
Build outline	Alt+U
Load data	Alt+L
Build outline and load data	Alt+O
Add to existing data	Alt+X
Subtract from existing data	Alt+S
Overwrite existing data	Alt+G
Delete all members first	Alt+M
Delete and restore database	Alt+R

Element Name	Equivalent
Incremental load	Alt+I
Create and save rule file only	Alt+C
Number of records to keep: All	Alt+A
Number of records to keep: Limit	Alt+T
Error file name: Default	Alt+D
Error file name: File name	Alt+E
Stop the data load when the first record is rejected	Alt+P
Deploy now	Alt+Y
Save as MaxL load script	Alt+V
Browse	Alt+W

Table 69 Keyboard Equivalents for **Cube Deployment Wizard**, **Incremental Load** Page

Element Name	Equivalent
Update all hierarchies	Alt+H
Update or rebuild selected hierarchies	Alt+S
Update	Alt+U
Rebuild	Alt+R
Clear	Alt+C
Preserve	Alt+P

Table 70 Keyboard Equivalents for Drill-through Report Editor, **Context Tab**

Element Name	Equivalent
Name	Alt+N
Description	Alt+D
Intersections	Alt+I
Add	Alt+A
Advanced Settings (applies to recursive hierarchies only)	Alt+G

Table 71 Keyboard Equivalents for Drill-through Report Editor, **Report Contents** Tab—**Report Type: Relational**

Element Name	Equivalent
Report Type: Relational	Alt+R

Element Name	Equivalent
Add	Alt+A
Show duplicate records in report	Alt+P
User user-defined SQL	Alt+D
Template SQL	Alt+Q
Filter	Alt+L
Row governor	Alt+V
Test	Alt+S

Table 72 Keyboard Equivalents for **Drill-through Template SQL** Dialog Box

Element Name	Equivalent
Cube Schema	Alt+S
Get Standard SQL	Alt+G
User-defined SQL	Alt+Q
Validate	Alt+V
Update User-defined SQL	Alt+U

Table 73 Keyboard Equivalents for **Drill-through Filter** Dialog Box

Element Name	Equivalent
Formula	Alt+F
Filter Expression	Alt+X

Table 74 Keyboard Equivalents for **Drill-through Report Testing** Dialog Box

Element Name	Equivalent
Cube Schema	Alt+S
Column value for each intersection	Alt+I
Show result	Alt+R
Maximum rows to display	Alt+M
Resulting report	Alt+T

Table 75 Keyboard Equivalents for Drill-through Report Editor, **Report Contents** Tab—**Report Type: URL**

Element Name	Equivalent
Report Type: URL	Alt+U
URL	Alt+L
Sample URL Template	Alt+S
FDM URL Template	Alt+F
OBI URL Template	Alt+O

Table 76 Keyboard Equivalents for Drill-through Report Editor , **Report Contents** Tab—**Report Type: Java Method**

Element Name	Equivalent
Report Type: Java Method	Alt+J
Java Class Name	Alt+C
Java Method Name	Alt+M

Table 77 Keyboard Equivalents for Drill-through Report Editor, **Associations** Tab

Element Name	Equivalent
Select all Essbase models	Alt+A
Refresh Model List	Alt+R

Table 78 Keyboard Equivalents for **Update Cube Linkage** and **Cube Linkage** Dialog Boxes

Element Name	Equivalent
Update all Essbase applications and databases to link to the current Studio Server (all Essbase instances must be started)	Alt+P
Stop all Essbase applications after the update	Alt+S
Update selected Essbase application and database	Alt+D
Update	Alt+U **Note:** You must first have selected an application and database, listed under **Essbase Connections** in the table.
Essbase Studio Server	Alt+S
Essbase model name	Alt+M

Table 79 Keyboard Equivalents for **Export** Dialog Box

Element Name	Equivalent
File Name	Alt+N

Element Name	Equivalent
File Description	**Alt**+**D**
Browse	**Alt**+**W**
File Contents	**Alt**+**C**
Export entire catalog	**Alt**+**E**
Export partial catalog (use the Add Button to select elements)	**Alt**+**P**
Add, then **Add Metadata**	**Alt**+A, then **M**
Add, then **Add Data Source**	**Alt**+**A**, then **D**
Include associated Essbase models with Cube Schemas	**Alt**+**I**

Table 80 Keyboard Equivalents for **Import** Dialog Box

Element Name	Equivalent
Browse	**Alt**+**W**
Check for elements and overwrite them	**Alt**+**C**
Check for elements but do not overwrite them	**Alt**+**K**
Do not check for elements; all elements are new	**Alt**+**E**

Table 81 Keyboard Equivalents for the **EIS Catalog Migration** Dialog Box

Element Name	Equivalent
EIS Catalog ODBC DSN	**Alt**+**E**
User	**Alt**+**U**
Password	**Alt**+**P**
Fetch Models	**Alt**+**F**
New Data Source	**Alt**+**N**
Browse	**Alt**+**B**
In **Select Folder** dialog box: **Create Folder**	**Alt**+**F**
Migrate	**Alt**+**M**

Table 82 Keyboard Equivalents for the **Find** Dialog Box

Element Name	Equivalent
Find	**Alt**+**F**
Match Case	**Alt**+**C**

Element Name	Equivalent
Match Word	**Alt+W**
Apply to current selection	**Alt+A**

Keyboard Equivalents in the Minischema Editor

The tables in this section describe keyboard equivalents used in the Essbase Studio minischema editors.

- Table 83, "Accessing Items on the Context Menu in a Minischema Diagram," on page 350
- Table 84, "Accessing Additional Minischema Context Menu Items When a Minischema Table is Selected," on page 350

Table 83 Accessing Items on the Context Menu in a Minischema Diagram

Action	Key or Key Combination
Activate the context menu.	**Shift+F10**
Save the active minischema.	**Shift+F10**, down arrow key once, then **Enter**
Launch the **Add Join** dialog box.	**Shift+F10**, then **J**
Launch the **Add Joins by Inspection** dialog box.	**Shift+F10**, then **I**
Launch the **Properties** dialog box for the minischema. (The **Properties** dialog contains one tab, the **Add/Remove Tables** tab.)	**Shift+F10**, then **A** (**Add/Remove Tables** command) Or **Shift+F10**, then **P** (**Properties** command) **Note:** Both of these commands access the same dialog box.
Lay out the minischema in the work area.	**Shift+F10**, then **L**
Zoom in on, or enlarge the view of, the minischema.	**Ctrl+=**
Zoom out on, or reduce the view of, the minischema	**Ctrl+-**

Table 84 Accessing Additional Minischema Context Menu Items When a Minischema Table is Selected

Action	Key or Key Combination
Move focus from one table to another in the minischema diagram	Arrow keys
Remove selected table or join	**Shift+F10**, then **R**
Minimize the selected table in the minischema diagram	**Shift+F10**, then **N**
Maximize the selected minimized table in the minischema diagram	**Shift+F10**, then **X**
Add a join to the selected table	**Shift+F10**, then **J**

Action	Key or Key Combination
Add joins by inspection to the selected table	**Shift**+**F10**, then **I**
Launch the **Edit Properties** dialog box for the minischema (this is the **Add/Remove Table** command on the context menu)	**Shift**+**F10**, then **A** Or **Shift**+**F10**, then **P**
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.	**Shift**+**F10**, then down arrow key to **Add to Metadata Navigator**
Display a drop-down list of colors, use arrow keys then press **Enter** to select a color	**Shift**+**F10**, then **C**
Automatically arrange, or lay out, the minischema in the work area.	**Shift**+**F10**, then **L**
Zoom in on, or enlarge the view of, the minischema.	**Ctrl**+**=**
Zoom out on, or reduce the view of, the minischema	**Ctrl**+**-**
Launch sample data in the work area for the selected table	**Shift**+**F10**, then **V**
Launch the Properties dialog box for the selected table	**Shift**+**F10**, then **P**

B Limitations and Guidelines

In This Appendix

Overview

This appendix describes limitations you may encounter while working with Essbase Studio.

Catalog and Data Sources Guidelines

The following topics describe limitations involving the Essbase Studio catalog, physical data sources, and data source connections:

- “One Essbase Studio Server Per Catalog Database” on page 354
- “Catalog Access Guidelines” on page 354
- “Catalog and Data Source Permission Guidelines” on page 354
- “Passwords Not Included in Catalog Export File” on page 355
- “Oracle Client Driver Guidelines” on page 355
- “MySQL Limitation” on page 355
- “Excel Files as Data Source Not Supported” on page 355
- “General Catalog and Data Source Limitations and Guidelines” on page 355

One Essbase Studio Server Per Catalog Database

The Essbase Studio catalog database should not be used by two or more Essbase Studio Server instances, either simultaneously or in succession. Oracle *strongly* recommends that each Essbase Studio Server point to its own unique catalog database.

Catalog Access Guidelines

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. Other requests from 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 37 and "server.writeLockTimeOut" on page 38 for more information.

You may also choose to unlock the catalog during the scraping process by selecting the "Lock catalog during exploration" check box in the **Select Tables** page of the **Connection Wizard**, as described in "Selecting Tables for Relational Sources" on page 77.

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 used for your Essbase Studio catalog database, Oracle recommends setting the following privileges for the user that is used to connect to catalog database.

  ```
  open_cursors=nnn SCOPE=MEMORY
  ```

  where `nnn` >= 300.
- 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.

Passwords Not Included in Catalog Export File

When exporting data source connections, the password to the source database is not included in the resulting XML file. After importing the XML file, add the password back by editing the connection, as described in "Editing Data Source Connection Properties" on page 99.

MySQL Limitation

In MySQL, changes made to database tables—for example, renaming columns—are not properly reflected in corresponding MySQL views. This causes unexpected results when refreshing MySQL data sources containing views in Essbase Studio Console.

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 and individual text file names must not contain spaces.
- Text file data source directory paths are limited in length as follows:
  - Native mode: 121 bytes
  - Unicode mode: 1028 bytes
- 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 data source names (DSNs).
- If source data is added, modified, or deleted while the sample data viewing window is open, data does not automatically refresh in the open window. You must close the window and reissue the **View Sample Data** command to view the changed data.

Introspection Limitations

- Introspection is not supported for these data sources:
  - Text files
  - Oracle BI EE

Metadata Elements Usage Guidelines and Limitations

The following topics describe limitations in the usage of metadata elements:

- "Derived Text Measures Limitations" on page 356
- "Cycle Dependency Guidelines" on page 356
- "Rules for Generating Key and Caption Bindings for Oracle BI EE Business Model Dimension Elements" on page 357

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 and **XOLAP Model with Oracle OBIEE** option will not be available for the model.
- Derived text measures are not fully supported in Spreadsheet Add-in. Use Smart View to view members in derived text measures.

Cycle Dependency Guidelines

Essbase Studio does not verify cycle dependencies between metadata elements. Cycle dependency is a relationship between metadata elements that cycles back on 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, which 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.

Rules for Generating Key and Caption Bindings for Oracle BI EE Business Model Dimension Elements

During exploration of Oracle BI EE Business Model data sources, Essbase Studio creates a hierarchy for each selected Oracle BI EE dimension. For each Oracle BI EE dimension level that contains at least one primary key column, Essbase Studio creates a dimension element representing the given level of the Oracle BI EE dimension.

Rules used to generate binding expressions for these dimension elements:

Key Binding Rules

- Rule 1—If the Oracle BI EE dimension level contains a single primary key, then the primary key column is a key binding for the dimension element.
- Rule 2—If the Oracle BI EE dimension level contains several primary keys, the key binding expression is the result of the concatenation of all primary keys (converted to `string` if its unit is not `string`) with an underscore (_) as a delimiter.

Note: If the Oracle BI EE dimension level does not contain a logical primary key column, then Essbase Studio does not create a dimension element for the given Oracle BI EE dimension level.

Caption Binding Rules

- Rule 1—If the unit of the key binding expression is `string` (not as a result from concatenation of several primary keys), then the caption binding for the dimension element is equal to the key binding expression.
- Rule 2—If the unit of the key binding expression is not `string`, then the first Oracle BI EE logical column level that is a key column (not a primary key column), and has type of CHAR or VARCHAR, is used as a caption binding expression for the given dimension element.
- Rule 3—If the unit of the key binding expression is not `string`, and there is no key column with type CHAR or VARCHAR, then the key binding conversion, `.toString`, is used as a caption binding expression for the given dimension element.
- Rule 4—If the unit of the key binding expression has unit `string` as a result of the concatenation of several primary keys, then the caption binding expression is the same as the key binding expression.

Hierarchies Usage Guidelines and Limitations

The following topics describe guidelines and limitations in using hierarchies:

- "Hierarchy Guidelines" on page 358
- "Standard and Measure Hierarchies Limitations" on page 358
- "Calendar Hierarchies Limitations" on page 358

Hierarchy Guidelines

These guidelines apply to standard and measure hierarchies:

- When working with relational data sources:
  - Columns in a hierarchy can be added from columns in the **Metadata Navigator** or the **Data Source Navigator**.
  - The columns in a single-chain hierarchy must come from one or more tables within a single data source.
  - In a multichain hierarchy, each chain in the hierarchy may come from different data sources; however, within a single chain in the multichain hierarchy, columns must come from one or more tables within a single data source.
  - When creating a cube schema using a multichain hierarchy built from multiple data sources, you must override the data load binding on the lowest level (Gen1 or leaf level) of each chain in the hierarchy, as described in "Defining Data Load Mappings" on page 179.
- When working with text file data sources, columns in a hierarchy—single-chain or multichain—must come from the same text file within a single data source.

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.
  - For deployment, Essbase Studio Server must be run in streaming mode (described in "server.essbase.streamingCubeBuilding" on page 34).
  - 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:

Essbase Model Rebuilding Guidelines

This topic outlines Essbase model handling of specific metadata element and cube schema changes, noting whether the changes require that a model be rebuilt or not.

Operations that do not require rebuilding Essbase model

Recreating or rebuilding an Essbase model *is not* required when you perform the following operations on a metadata folder, dimension element, derived text measure, text list, hierarchy, measure hierarchy, or cube schema:

- Rename
- Move

Further, recreating or rebuilding an Essbase model *is not* required when you perform these operations:

- Change the binding, filter, sort order, or alias set bindings of a dimension element
- Change the binding, range, or alias set bindings of a derived text measure
- Change the value binding or ID binding of a text list
- Change an overridden data load binding in a cube schema
- For models that contain varying attributes:

- ❍ Rename the hierarchy that has a varying attribute binding
- ❍ Rename the dimension element that is used as the leaf level in a varying attribute dimension
- ❍ Rename the hierarchy that is the independent dimension in the varying attribute

Operations that require rebuilding an Essbase model

Note that recreating or rebuilding an Essbase model *is* required when you perform the following operations:

- Modify hierarchy content; for example, reorder, add, or remove members in a hierarchy or measure hierarchy
- Change the dimensionality in a cube schema; for example, add or remove hierarchies
- Add or remove any loose measures in a cube schema
- Change the measure hierarchy in a cube schema
- Override the default data load bindings in a cube schema

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.

Duplicate Member Name Support Limitation

Essbase models that are enabled for duplicate member support will have a validation error when the model contains a non-string dimension element with a delayed key binding. Essbase Studio Server cannot generate the concatenated key binding required at deployment time.

To work around this, you can clear the "Duplicate member name support" option in the Essbase model. But if this is not feasible, these are possible solutions:

- Set the key binding in the dimension element that contains the delayed key binding.

  See "Choosing a Key Binding Option for a Dimension Element" on page 137.
- Set the key binding for the dimension element in the hierarchy where it is used.

  See "Using Delayed Key Bindings in Hierarchies" on page 159.

XOLAP Functionality Guidelines

See also "Guidelines for Using XOLAP" on page 190.

These guidelines apply to XOLAP functionality:

- Alternate hierarchies are supported by Essbase Studio when deploying an Essbase model enabled for XOLAP if the join paths for all chains come from the same set of dimension tables. The following example alternate hierarchies use the TBC database to illustrate.
  - In the following example, the join path to the fact table comes from the same dimension table.

    ```
    - PKGTYPE               (TBC.PRODUCT)
       - SKU                (TBC.PRODUCT)
    - OUNCES                (TBC.PRODUCT)
       - SKU_ALIAS          (TBC.PRODUCT)
    ```

    In this case, the join path to the SALES fact table is:

    TBC.PRODUCT joins TBC.SALES.
  - In the following example, the join path to the fact table comes from the same set of dimension tables.

    ```
    - FAMILY                (TBC.FAMILY)
       - SKU                (TBC.PRODUCT)
    - FAMILYID              (TBC.FAMILY)
       - SKU_ALIAS          (TBC.PRODUCT)
    ```

    In this case, the join path to the SALES fact table is:

    TBC.FAMILY joins TBC.PRODUCT joins TBC.SALES.
  - In the following example, the join path to the fact table comes from different dimension tables.

    ```
    - UDAMKTTYPE            (TBC.MARKET)
       - STATE              (TBC.MARKET)
    - PKGTYPE               (TBC.PRODUCT)
       - SKU                (TBC.PRODUCT)
    ```

    In this case, the alternate hierarchy scenario is not supported because there are two distinct join paths to the SALES fact table:

    1. TBC.MARKET joins TBC.SALES

    2. TBC.PRODUCT joins TBC .SALES
  - In the following example, the join path to the fact table comes from different sets of dimension tables.

    ```
    - FAMILYID              (TBC.FAMILY)
       - INTRODATE          (TBC.PRODUCT)
    - SKU_ALIAS             (TBC.PRODUCTDIM)
       - CAFFEINATED        (TBC.PRODUCT)
    ```

    In this case, the alternate hierarchy scenario is not supported because there are two distinct join paths to the SALES fact table:

    1. TBC.FAMILY joins TBC.PRODUCT joins TBC.SALES

    2. TBC.PRODUCTDIM joins TBC.PRODUCT joins TBC .SALES
- You may tag a dimension with alternate hierarchies as **Multiple Hierarchy Enabled** in the **Essbase Model Properties** dialog box, **Outline Build** tab).

- Essbase Studio can create attribute dimensions in Essbase models enabled for XOLAP; however, attribute dimensions must have only one child level.

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.

Varying Attribute Editing Guidelines

The following scenarios necessitate redefining or editing varying attributes:

- After performing a catalog export/import, the varying attribute information in Essbase models will be lost. You must redefine all varying attribute members in any Essbase models that contain them after catalog export/import.
- If you rename a data source connection that is used in a model, any varying attribute members in the model that reference the old connection name will be out of synch. You must edit the varying attribute members to reference the new data source connection name. In the **Essbase Model Properties** dialog box, locate the varying attribute members and access the **Edit Varying Attributes** dialog box. In the **Independent Dimension Settings** group:
  - If the varying attribute type is “Individual”, edit the expression in the **From** column.
  - If the varying attribute type is “Range”, edit the expressions in the **From** and **To** columns.

See “Setting Varying Attributes for Members” on page 222 for information on creating and editing varying attributes.

Text File Data Source Member Transformation Limitation

If your text file data source contains double quotes (`""`), ignore them when creating transformation rules. The double quotes will be removed by Essbase during the cube deployment process. For example, for a text value `"2000-01-01"` where double quotes are included, you should use `substr(1,4)` to derive the year value of 2000.

Cube Deployment Limitations and Guidelines

The following topics describe limitations and guidelines involving cube deployment:

- "General Limitations" on page 363
- "Rules File Limitations and Guidelines" on page 364

General Limitations

These general 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.
  - Essbase Studio Server must be run in streaming mode (described in "server.essbase.streamingCubeBuilding" on page 34).
  - In the **Load data options** group, selecting the **Overwrite existing data** option is not allowed.
- If you add data load bindings to a cube schema from which an Essbase model was created, the cube schema and model will be out of sync. If you deploy from the existing Essbase model, the model does 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.
- Incremental builds for XOLAP-enabled models are not supported.

  If you are deploying from an Essbase model that is enabled for XOLAP, it is highly recommended that you use a new application and database name; or use the **Delete all members first** option to deploy over an existing XOLAP application.
- You cannot deploy an XOLAP-enabled Essbase model that is based on an Oracle BI EE data source.
- In nonstreaming mode (`server.essbase.streamingCubeBuilding=false` in `server.properties`), Essbase Studio can deploy cubes only from Oracle BI EE data sources version 10.1.3.4 or later. Cubes may be deployed from an earlier version of Oracle Business Intelligence Enterprise Edition, 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 34 for information on this property.
- **IBM DB2:** A limitation in IBM DB2 handling of the LONG VARCHAR data type in a select DISTINCT statement causes cube deployment to fail. To avoid this failure, set `server.essbase.disableDistinct` to true.

  See "server.essbase.disableDistinct" on page 39 for information on setting this property.
- If a data source column is bound to a level-0 member set, and contains both base members and shared members, it is possible that the data aggregated in the Essbase Studio-generated data load SQL for the specific base members (which have shared members in the same dimension) will be augmented at times, depending on the number of shared members. In

this case, Oracle recommends that users define their own custom data load SQL, as described in "Overriding Standard Data Load SQL" on page 187.

For example, using the TBC sample, the data aggregated for 100-10 in the standard data load SQL is twice as much as it should be (doubled) because 100-10 has shared members under "Diet". Users can manually add one more "group by" column, "Family", to the custom data load SQL and select the **Use Custom SQL for data load** option to get the correct aggregated data.

Rules File Limitations and Guidelines

In Administration Services Console, when working with rules files generated by Essbase Studio, these guidelines apply:

- Rules files for data loads can be previewed only; no editing is allowed.
- Rules files for dimension builds (member loads) cannot be opened or previewed.
- If a measures hierarchy exceeds 250 members, Essbase Studio creates one rules file for every 250 members.

Drill-through Reports Limitations and Guidelines

The following topics describe limitations and guidelines involving drill-through reports:

- "Guidelines for Accessing Drill-through Reports in Spreadsheet Add-in" on page 364
- "General Drill-through Operations Limitations" on page 365

Guidelines for Accessing Drill-through Reports in Spreadsheet Add-in

Smart View is the preferred spreadsheet client for performing drill-through on cubes created in Essbase Studio Server. When using Spreadsheet Add-in to perform drill-through, note the following limitations:

- 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.
- When performing drill-through on cubes created from Essbase Studio, Spreadsheet Add-in does not support localization. To clarify, for Essbase Studio-built cubes, if the relational data source code page is not English, then drill-through does not work. This means that from Spreadsheet Add-in, you cannot drill through to non-English sources.

  Use Smart View to access non-English sources.

- 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 Smart View, a message is returned stating that Essbase Studio Server is not running.
- In 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

- 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.
- With Essbase Studio-built cubes, drill-through on a range of cells is not available for Oracle Hyperion Smart View for Office, Fusion Edition or Oracle Essbase Spreadsheet Add-in.

C Naming Restrictions for Essbase Studio

In This Appendix

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 85.

Table 85 List of Restricted Characters in Application and Database Names

Character	Description
*	asterisk
[]	brackets
:	colon
;	semicolon
,	comma
=	equal sign
>	greater-than sign
<	less-than sign
.	period
+	plus sign
?	question mark

Character	Description
"	double quotation mark
‘	single quotation mark
/	forward slash
\	backslash
|	vertical bars
	tabs

- 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 Essbase Naming Conventions” on page 368
- “Metadata Element Naming Restrictions” on page 369

Metadata Elements Subject to Essbase Naming Conventions

Essbase naming conventions apply to the following metadata elements:

- All hierarchy types
- Dimension elements

  Note: Any dimension element that is not column based; that is, binding in a string constant, is subject to Essbase naming conventions.

- Derived text measures
- Alias set names

- Drill-through reports

Metadata Element Naming Restrictions

The following topics list naming restrictions as they apply to metadata elements:

- "General Metadata Element Naming Guidelines" on page 369
- "Restricted Characters" on page 369
- "Reserved Words" on page 370

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 372.

- 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 86:

Table 86 List of Restricted Characters for Metadata Element Names

<table>
<tr><th>Character</th><th>Description</th></tr>
<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><</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>vertical bar</td></tr>
</table>

Reserved Words

You may use reserved words when naming the following:

- Essbase cubes
- Cube schemas
- Essbase models
- Metadata elements

Do not use any of the following words in the names of hierarchies or user-defined members.

ALL
AND
ASSIGN
AVERAGE
CALC
CALCMBR
COPYFORWARD
CROSSDIM
CURMBRNAME
DIM
DIMNAME

DIV
DYNAMIC
EMPTYPARM
EQ
EQOP
EXCEPT
EXP
EXPERROR
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
$$$UNIVERSE$$$
#MISSING
#MI

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

D CPL Reference

In This Appendix

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 comprise:

- "Operands" on page 373
- "Operators" on page 377

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 indicates whether 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'\
  ```

- `'<NAMEx>'`

  Element names must be in single quotation marks. 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, which indicates a class, 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:

  `1000000000000001L`

- **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. String constants are expressed in these ways:

  - With quotes:

    `"Example of string value"`

    If you want double quotation marks inside a string, you must double the double quotation mark character within the string, for example:

    `"Example of string value "" with quote inside"`

- With square brackets:

```
$[[Example of string value " with quote inside]]$
```

- With parentheses:

```
$((Example of string value " with quote inside))$
```

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 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 87. Priority 1 is the highest priority; priority 8 is the lowest.

Table 87 Operator Priority

Priority	Operators
1	`( )`
2	`not  +  - (Unary)`
3	`*  /  mod`
4	`+  - (Binary)`
5	`||`
6	`<  <=  >  >=  ==  !=`
7	`and`
8	`or`

Operands with the same priority are executed from left to right.

Grouping

You use grouping operators to associate subexpressions.

`()` Grouping—parentheses

Groupings in parentheses override the standard operator level of precedence. They 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

```
(connection :
  \'DB_conn'::'TBC.tbc.SALESFACT'.'OPENINGINVENTORY')
```

With Connections

```
(
  (connection :
    \'DB_conn'::'TBC.tbc.SALESFACT'.'OPENINGINVENTORY')
-
  (connection :
    \'DB_conn'::'TBC.tbc.SALESFACT'.'SALES')
)
*
connection : \'DB_conn'::'TBC.tbc.SALESFACT'.'COGS'
```

Logical Operators

You use logical operators to evaluate conditions.

`not` Boolean—unary not.

Returns:

- True—when a condition is false
- False—when a condition is true

Example:

```
not true
```

`and` Boolean—binary and. Commutative.

Returns:

- True—if both conditions are true
- False—if either condition is false

Example:

```
true and false
```

`or` Boolean—binary or. Commutative.

Returns:

- True—if either condition is 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

```
connection :
  \'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—noncommutative.

Example:

```
100 / 10
```

`mod` Modulus (integer)—returns the remainder of two divisibles.

Example:

```
18 mod 5
```

returns 3.

Examples

With Connections

```
(connection :
  \'DB_conn'::'TBC.tbc.SALESFACT'.'SALES')
-
(connection :
  \'DB_conn'::'TBC.tbc.SALESFACT'.'COGS')
```

With Grouping Operators

```
(
  (connection :
    \'DB_conn'::'TBC.tbc.SALESFACT'.'OPENINGINVENTORY')
-
  (connection :
    \'DB_conn'::'TBC.tbc.SALESFACT'.'SALES')
)
*
connection : \'DB_conn'::'TBC.tbc.SALESFACT'.'COGS'
```

String Operators

You use string operators on strings.

`||` Concatentation

Concatentation can be applied only to string expressions or sets. Combines multiple strings into one. Is noncommutative.

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:

```
"A  s t r i n g"
 | | | | | | | |
 1 2 3 4 5 6 7 8
```

contains

Determines whether 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 the substring is not in the string

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 lowercase.

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 uppercase.

Syntax

```
upper ( <string> )
```

Parameters

- `<string>` is a string.

Returns

String

Example

```
upper ( connection : \'DB'::'TBC.tbc.MARKET'.'STATE' )
```

Numeric

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

```
log10 ( <number> )
```

Parameters

- `<number>` is a number.

Returns

Number

Example

```
log10 ( 136 )
```

pow

Returns the value of x to the power of y.

Syntax

```
pow ( <number_x> , <number_y> )
```

Parameters

- `<number_x>` is a number.
- `<number_y>` is a number.

Returns

Number

Example

```
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 )
```

Glossary

! See bang character.

#MISSING See missing data.

access permissions A set of operations that a user can perform on a resource.

accessor Input and output data specifications for data-mining algorithms.

accounts dimension A dimension type that makes accounting intelligence available. Only one dimension can be defined as Accounts.

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 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 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.

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.

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 requirements, reporting requirements, or both.

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.

attribute A 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.

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.

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.

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.

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 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.

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.

cache A buffer in memory that holds data temporarily.

calculated member in MaxL DML A member designed for analytical purposes and defined in the optional WITH section of a MaxL DML query.

cascade The process of creating multiple reports for a subset of member values.

CDF See custom-defined function.

CDM See custom-defined macro.

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.

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

clean block A data block in which 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.

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.

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.

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.

crosstab reporting Reporting that 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 by the dollar (1* 0.923702). After conversion, the European euro amount is .92.

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-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.

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.

cycle through Perform multiple passes through a database while calculating it.

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 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 mining The process of searching through an Essbase database for hidden relationships and patterns in a large amount of data.

data value See cell.

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 analysis types 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.

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.

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.

dimension A data category used to organize business data for the 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 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) }

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, if they have been updated).

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.

duplicate alias name A name that occurs more than once in an alias table and 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 Multiple occurrences 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 Members 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 Members 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 reference A pointer in the rules file to header records in a data source.

Dynamic Time Series A process that performs period-to-date reporting in block storage databases.

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

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.

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.

external authentication Logging on to Oracle EPM System products with user information stored outside the application. The user account is maintained by the EPM System, but password administration and user authentication are performed by an external service, using a corporate directory such as Oracle Internet Directory (OID) or Microsoft Active Directory (MSAD).

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.

file delimiter A character, such as a comma or tab, that separates 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.

format string 1) In Essbase, a method for transforming the way cell values are displayed. 2) In FDM, a parameter of a Format or Formatted Date derived property that indicates the format in which a property value should be returned.

free-form reporting Creating reports by entering dimension members or report script commands in worksheets.

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.

global report command A command in a running report script that is effective until it is replaced by another global command or the file ends.

GUI Graphical user interface

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.

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. An index page contains pointers to data blocks.

input data Data loaded from a source rather than calculated.

intelligent calculation A calculation method tracking updated data blocks since the last calculation.

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 nonintersecting dimensions, the nonintersecting dimensions are not relevant to that dimension.

introspection A deep inspection of a data source to discover hierarchies based on the inherent relationships in the database. Contrast with scraping.

isolation level An Essbase Kernel setting that determines the lock and commit behavior of database operations. Choices are: committed access and uncommitted access.

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 same layer, though they are in the same generation.

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.

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 taskflow, the point where the activity in one stage ends and another begins.

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.

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, 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.

Log Analyzer An Administration Services feature that enables filtering, searching, and analysis of Essbase logs.

LRO See linked reporting object.

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 The data definition language used by Essbase for batch or interactive system-administration tasks.

MaxL DML The 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).

member A discrete component within a dimension. A member identifies and differentiates the organization of similar units. For example, a time dimension might include members Jan, Feb, and Qtr1.

member load In Essbase Integration Services, 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.

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

mining attribute In data mining, a class of values used as a factor in analysis of a set of data.

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 nondimensional 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.

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.

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.

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.

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.

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.

OLAP Metadata Catalog In Essbase Integration Services, a relational database containing metadata describing the nature, source, location, and type of data that is pulled from the relational data source.

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

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.

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.

page file An 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.

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.

partition area A subcube 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.

periodicity Any shared pattern among time-related dimension members that makes them meaningful for time-based analysis. For example, Jan and Apr share the periodicity of being opening months of a quarter in the Gregorian calendar.

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.

pivot 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.

precalculation Calculating the database before user retrieval.

preserve formulas User-created formulas kept within a worksheet while retrieving data.

provisioning The process of granting users and groups specific access permissions to resources.

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 governor An Essbase Integration Server parameter or Essbase Server configuration setting that controls the duration and size of queries made to data sources.

ragged hierarchy An asymmetrical hierarchy that occurs when a member has branches that contain different numbers of levels. For example, assume a Country dimension containing hierarchies of different geographical entities down to cities at level 0. The United States hierarchy contains three levels: country, state, and city. The Greece hierarchy contains two levels: country and city.

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.

redundant data Duplicate data blocks that Essbase retains during transactions until Essbase commits updated blocks.

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 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.

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.

roll-up See consolidation.

root member The highest member in a dimension branch.

sampling The process of selecting a representative portion of an entity to determine the entity's characteristics. See also metadata sampling.

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

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. The levels, from highest to lowest: the entire system (Essbase Server), applications on Essbase Server, or databases within Essbase Server applications. See also persistence.

scraping An inspection of a data source to derive the most basic metadata elements from it. Contrast with introspection.

security platform A framework enabling Oracle 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.

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 The part of the Shared Services repository that manages EPM System deployment information for most EPM System products, including installation directories, database settings, computer names, ports, servers, URLs, and dependent service data.

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 (SSO) The ability to log on once and then access multiple applications without being prompted again for authentication.

slicer In MaxL DML, the section at the end of a query that begins with and includes the keyword WHERE.

sparse dimension In block storage databases, a dimension unlikely to contain data for all member combinations when compared to other dimensions. Contrast with dense dimension. For example, not all customers have data for all products.

standard dimension A dimension that is not an attribute dimension.

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.

supervisor A user with full access to all applications, databases, related files, and security mechanisms for a server.

suppress rows A setting that 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 simultaneously connect to an single instance.

synchronized The condition that exists when the latest version of a model resides in both the application and in Shared Services. See also model.

TCP/IP See Transmission Control Protocol/Internet Protocol.

text list In Essbase, an object that stores text values mapped to numeric identifiers. Text Lists enable the use of text measures.

text measure 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 series reporting A process for reporting data based on a calendar date (for example, year, quarter, month, or week).

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, measures, and, ultimately, to its physical source elements. See also lineage.

transformation 1) A process that transforms artifacts so that they function properly in the destination environment after application migration. 2) In data mining, the modification of data (bidirectionally) flowing between the cells in the cube and the algorithm.

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 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.

triggers An Essbase feature whereby data is monitored according to user-specified criteria that, 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.

unique member name A nonshared member name that exists only once in a database outline.

unique member outline A database outline that is not enabled for duplicate member names.

upper-level block A type of data block wherein at least one of the sparse members is a parent-level member.

user-defined attribute (UDA) An attribute, associated with members of an outline to describe a characteristic of the members, that can be used to return lists of members that have the specified associated UDA.

validation The process of checking a business rule, report script, or partition definition against the outline to ensure that the object being checked is valid.

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.

visual cue A formatted style, such as a font or a color, that highlights specific data value types. 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.

WITH section In MaxL DML, an optional section of the query used for creating reusable 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.

workbook An entire spreadsheet file with many worksheets.

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

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.

Index

D

E

F

N

O

P

Q

R

S

T

U

V

W