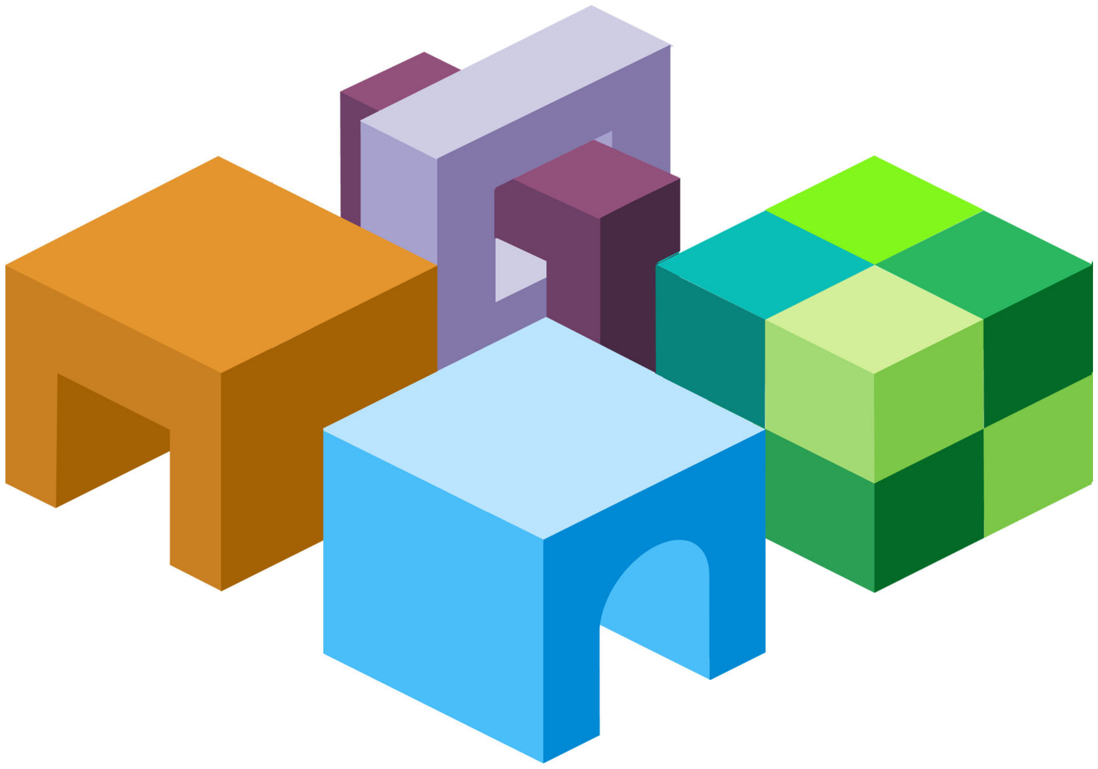


ORACLE® ESSBASE

RELEASE 11.1.1

DATABASE ADMINISTRATOR'S GUIDE

ORACLE®
ENTERPRISE PERFORMANCE MANAGEMENT SYSTEM

Essbase Database Administrator’s Guide, 11.1.1

Copyright © 1996, 2008, Oracle and/or its affiliates. All rights reserved.

Authors: EPM Information Development Team

Contents

Part I

Understanding Essbase

In Understanding Essbase:

1 Introducing Essbase

In This Chapter

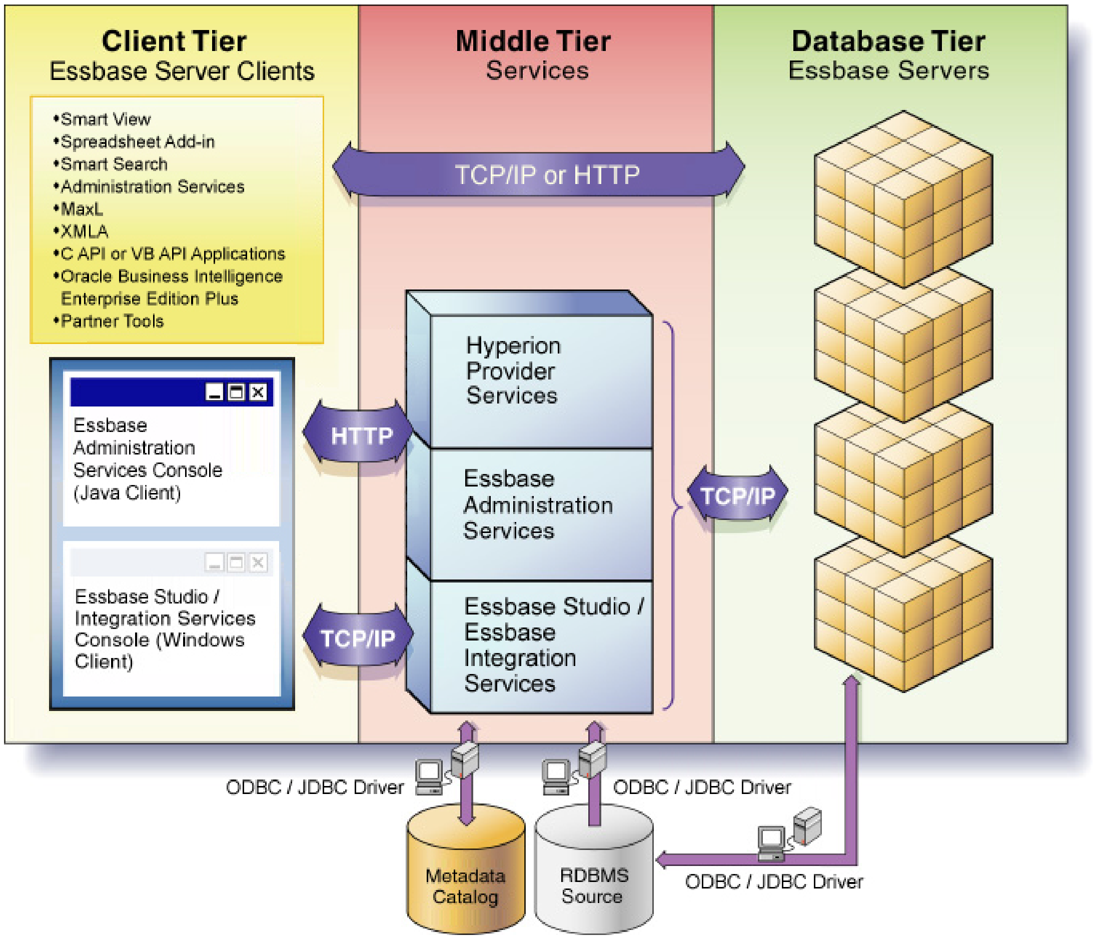
Introduction

Oracle Essbase products provide companies the ability to deliver critical business information to the right people when they need it. With Essbase, companies quickly leverage and integrate data from multiple existing data sources and distribute filtered information to end-user communities in the format that best meets the users' needs. Users interact and intuitively explore data in real time and along familiar business dimensions, enabling them to perform speed-of-thought analytics.

Essbase Product Components

Essbase products incorporate powerful architectural features to handle a wide range of analytic applications across large multi-user environments. Figure 1 provides a high-level view of the information flow between the three tiers of the Essbase architecture. The client tier (on the left) includes Essbase Server clients, such as Oracle Hyperion Smart View for Office, Fusion Edition, Administration Services Console and Oracle Hyperion Smart Search, Fusion Edition. The middle tier (in the center) includes services, such as Oracle Hyperion Provider Services and Oracle Essbase Administration Services. The database tier (on the right) is made up of Essbase Servers. Communication between the client and middle tiers, and the middle and database tiers, is through HTTP. Communication between the client and database tiers is through TCP/IP or HTTP. Communication between data sources and the metadata catalog with the middle and database tiers is through ODBC and JDBC drivers.

Figure 1 High-level Information Flow Between Product Components


Client Tier
Essbase Server Clients
•Smart View
•Spreadsheet Add-in
•Smart Search
•Administration Services
•MaxL
•XMLA
•C API or VB API Applications
•Oracle Business Intelligence Enterprise Edition Plus
•Partner Tools
Essbase Administration Services Console (Java Client)
Essbase Studio / Integration Services Console (Windows Client)
Middle Tier
Services
TCP/IP or HTTP
HTTP
TCP/IP
Hyperion Provider Services
Essbase Administration Services
Essbase Studio / Essbase Integration Services
TCP/IP
Database Tier
Essbase Servers
ODBC / JDBC Driver
ODBC / JDBC Driver
ODBC / JDBC Driver
Metadata Catalog
RDBMS Source

Essbase

Essbase—a multi-threaded OLAP database software that takes advantage of symmetric multiprocessing hardware platforms—is based on Web-deployable, thin-client architecture. The server acts as a shared resource, handling all data storage, caching, calculations, and data security. The Essbase Server client needs only to retrieve and view data that resides on a server.

All Essbase application components, including database outlines and calculation scripts, application control, and multidimensional database information, reside on a server. With Essbase, you can configure server disk storage to span multiple disk drives, enabling you to store large databases. Essbase requires a server to run a multi-threaded operating system so a server can efficiently manage simultaneous requests. A server also runs a server agent process that acts as a traffic coordinator for all user requests to applications.

Aggregate storage databases provide an alternative to block storage databases and enable dramatic increases in database dimensionality. Using aggregate storage, Essbase serves a wide range of analytic needs—financial analysis, planning, budgeting, sales analysis, marketing analysis, supply-chain analysis, profitability analytics—all from a single analytic infrastructure.

MaxL—a multidimensional database access language that is part of Essbase Server—provides a flexible way to automate Essbase administration and maintenance tasks.

For information on system requirements, see the *Oracle Hyperion Enterprise Performance Management System Installation Start Here*.

To install and configure Essbase, see the *Oracle Hyperion Enterprise Performance Management System Installation and Configuration Guide*.

Sample Essbase Applications

Essbase provides a set of sample applications and associated databases that you can use to learn about Essbase features, and which are the basis for many of the examples given in this document.

Sample applications are installed with Essbase Server.

To use the sample applications, see Appendix B, "Setting Up Sample Applications."

Administration Services

Administration Services—the database and system administrators' interface to Essbase—provides a single-point-of-access console to multiple Essbase Servers. Using Administration Services, you can design, develop, maintain, and manage multiple Essbase Servers, applications, and databases. You can preview data from within the console without having to open a client application, such as Spreadsheet Add-in. You can also use custom Java plug-ins to leverage and extend key functionality.

Essbase Studio

Essbase Studio simplifies cube construction by delivering a single environment for performing tasks related to data modeling, cube designing, and analytic application construction. With a wizard-driven user interface, Essbase Studio supports modeling of the various data source types from which Essbase applications are typically built.

A single 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 makes Essbase Studio inherently aware of the common metadata that is shared across the various applications enterprise wide.

Essbase Studio supports several drill-through options: relational databases, OBIEE, BI+ reports, URLs, Essbase applications (drill across), custom SQL, and Java methods. Essbase Studio also supports lineage tracking through a rich graphical view of metadata relationships, allowing users to follow application lineages to their metadata components and through to the data sources from which they were sourced.

Spreadsheet Add-in

Spreadsheet Add-in integrates Essbase with Microsoft Excel. Spreadsheet Add-in adds the Essbase menu to Excel, which provides enhanced commands such as Connect, Pivot, Drill-down, and Calculate. Users can access and analyze data on Essbase Server with mouse clicks and dragging operations. Spreadsheet Add-in enables multiple users to access and update data on an Essbase Server simultaneously.

Visual Explorer

Oracle Essbase Visual Explorer provides a query and analysis interface for creating interactive summaries and reports about the data in an Essbase database. You can choose to visualize data in various graphical formats, including bars, lines, Gantt bars, shapes, colors, and tables.

Visual Explorer is installed with and accessed from Spreadsheet Add-in.

Data Mining

Data Mining reveals hidden relationships and patterns in your data, enabling you to make better business decisions. Using Data Mining, you can plug in various data mining algorithms, build models, and apply them to existing Essbase applications and databases.

Integration Services

Oracle Essbase Integration Services—an optional product component—provides a metadata-driven environment to bridge the gap between data stored in Essbase databases and detailed data stored in relational databases. The Hybrid Analysis feature gives business users more detail for decision-making and IT managers more modularity in designing and maintaining large-scale analytic applications. Hybrid Analysis allows portions of Essbase databases to be stored in a relational database. This relational-stored data is mapped to the appropriate Essbase hierarchies.

Provider Services

Provider Services is a middle-tier data-source provider to Essbase for Java API, Smart View, and XMLA clients. Provider Services supports highly concurrent analytical scenarios and provides scalability and reliability in a distributed Web-enabled enterprise environment.

Smart View

Smart View provides a common Microsoft Office interface for Essbase, Oracle Hyperion Financial Management, Fusion Edition, Oracle Hyperion Planning, Fusion Edition, and Oracle Enterprise Performance Management Workspace, Fusion Edition data. Using Smart View, you can view, import, manipulate, distribute, and share data in Microsoft Excel, Word, and PowerPoint interfaces.

Application Programming Interface (API)

Essbase API—the developers' interface to Essbase—enables you to create customized applications. The *Oracle Essbase API Reference* provides a complete listing of API functions, platforms, and supported compilers.

Developer Products

Essbase developer products enable the rapid creation, management, and deployment of tailored enterprise analytic applications, whether or not users have programming knowledge.

The products (for example, Application Builder and Oracle's Hyperion® Application Builder for .NET) provide a comprehensive set of application programming interfaces, drag-and-drop components, and services.

Smart Search

Smart Search integrates with leading enterprise search solutions such as Google Search Appliance and Oracle Secure Enterprise Search to provide a familiar search interface. Using simple business terminology, users can obtain structured information from Essbase applications and databases. Information that has been filtered according to user privileges is delivered in data grids and live links in Smart View. Oracle Hyperion Smart Search, Fusion Edition greatly enhances the way in which users can quickly get to information contained within Oracle applications.

EPM System Lifecycle Management

Lifecycle Management provides a consistent way for Oracle Hyperion Enterprise Performance Management System products to migrate an application, a repository, or individual artifacts across product environments and operating systems. Generally, the Lifecycle Management interface in Oracle's Hyperion® Shared Services Console is consistent for all EPM System products that support Lifecycle Management. However, EPM System products display different artifact listings and export and import options in the Lifecycle Management interface.

Lifecycle Management features:

- Viewing applications and folders
- Searching for artifacts
- Comparing applications and folders
- Migrating directly from one application to another
- Migrating to and from the file system
- Saving and loading migration definition files
- Viewing selected artifacts
- Auditing migrations
- Viewing the status of migrations

- Importing and exporting individual artifacts for quick changes on the file system

In addition to providing the Lifecycle Management interface in Shared Services Console, there is a command-line utility called Lifecycle Management Utility that provides an alternate way to migrate artifacts from source to destination. The Lifecycle Management Utility can be used with a third-party scheduling service such as Windows Task Scheduler or Oracle Enterprise Manager.

Lastly, there is a Lifecycle Management Application Programming Interface (API) that enables users to customize and extend the Lifecycle Management functionality.

For detailed information about Lifecycle Management, see the *Oracle Hyperion Enterprise Performance Management System Lifecycle Management Guide*.

Key Features

Integration with Existing Infrastructure

Essbase products integrate with your existing business intelligence infrastructure. Essbase products meet the enterprise analytic demands of users for critical business information with a minimum of information technology (IT) overhead, and therefore enable organizations to realize maximum return on their existing IT investments:

- Provides an extensible architecture
- Supports a comprehensive range of data sources, hardware and operating system platforms, access interfaces, and development languages
- Enables analytic applications to be deployed across a local or wide area network and across an intranet or Internet

Data Integration

Essbase products enable organizations to leverage data in their data warehouses, legacy systems, online transaction processing (OLTP) systems, enterprise resource planning (ERP) systems, e-business systems, customer relationship management (CRM) applications, Web log files and other external data sources. For database integration, Integration Services provides a suite of graphical tools, data integration services, and a metadata catalog that tie into relational databases or data warehouse environments.

Ease of Server and Database Administration

Essbase products provide a cross-platform administration console. The console gives you detailed control over the Essbase environment:

- You can manage multiple servers and databases.
- You can use MaxL, a syntactical language command shell with a PERL extension module, to automate batch maintenance.

Mission Critical Applications in Web-based Environments

A middle-tier framework extends the power of Essbase products by creating a Web-enabled, distributed platform for Essbase applications, hence serving the analysis needs of large numbers of users in Web-based environments. Provider Services provides clustering and failover support, extending the scalability and reliability of the platform, and supports mission-critical applications in a 24 x 7 environment.

Powerful Querying

Large communities of business users can interact with data in real time to quickly analyze business performance. Using Essbase products, you can organize and present data along familiar business dimensions, enabling users to view and explore the data intuitively and turn it into actionable information.

Calculations

Essbase includes powerful calculation features for demanding analytic requirements. A rich library of functions makes it easy to define advanced and sophisticated business logic and relationships. Essbase gives users the flexibility to build, customize, and extend the calculator through custom-defined macros and functions, as well as the ability to span calculations across databases. On multiprocessor systems, a DBA can configure a single calculation request to use multiple threads to accomplish the calculation, providing enhanced calculation speed.

Aggregate storage databases provide an alternative to block storage databases and enable dramatic improvements in database aggregation time for certain types of applications.

Write-Back and Security

Essbase provides unique multi-user read and write capabilities, including data update and multi-user recalculation. Business users with front-end tools can write data back to a server and recalculate the data on a server using calculation scripts—key functionality to support sophisticated modeling and planning applications.

The robust, multilevel security model provides server-, database-, and cell-level security. Full control of data access, views, and write capabilities are managed through administration. Oracle's Hyperion® Shared Services provides integration with external authentication systems, such as Lightweight Directory Access Protocol (LDAP).

Ease of Development

Essbase offers many key advantages to help users develop effective multidimensional applications. Users can:

- Design and manage applications using a graphical interface to control most server functions.

- Quickly add dimensions, change calculations, and modify hierarchies to reflect new business developments. In addition, the dynamic dimension builder automatically defines and dynamically loads large amounts of data, including data from spreadsheets, flat files, and supported relational database tables directly into a database.
- Define key calculations without having to write a program.
- Define security for individuals and groups and customize views and retrieval procedures for each user without writing a program.

2 Quick Start for Implementing Essbase

Some information in this chapter applies only to block storage databases and is not relevant to aggregate storage databases.

Also see:

- Chapter 57, "Comparison of Aggregate and Block Storage"
- *Hyperion EPM System Upgrade Guide*, if you are upgrading from a previous version of Essbase

The following table provides a process map for implementing Essbase. Each step in the process is described at a high level, with references to where you can obtain more detailed information.

Table 1 A Process Map

Process Step	**Reference**
Learn the fundamentals of Essbase and distributed OLAP.	● Chapter 1, "Introducing Essbase" ● Chapter 3, "Understanding Multidimensional Databases" ● Chapter 6, "Creating Applications and Databases" ● Chapter 10, "Working with Attributes" ● Chapter 14, "Designing Partitioned Applications" ● Attend an Essbase training class; contact your software vendor.
Assess your needs and requirements. Have a clear idea of your data analysis needs and which calculations and reports you want to run.	Your budget, forecasting, and other financial reports with notes on how you want to improve them
Analyze your data from a multidimensional perspective: ● Where are your data sources? ● What type is the data? Is it detailed, relational data, or is it higher-level, hierarchical data that can be used for analysis? ● In what format is the data? ● How will you access the data? If you must access relational data, you may need Oracle Essbase SQL Interface or Integration Services.	● Chapter 3, "Understanding Multidimensional Databases" ● *Oracle Essbase SQL Interface Guide* ● ODBC drivers documentation ● Integration Services documentation
Install Essbase.	*Oracle Hyperion Enterprise Performance Management System Installation and Configuration Guide*
Design your application and database. ● Identify business and user requirements, including security.	Chapter 4, "Case Study: Designing a Single-Server, Multidimensional Database"

Process Step	Reference
• Identify source data and determine the scope of the Essbase database. • Choose whether to leave lowest-level member data in a relational database and access with Hybrid Analysis, or to load all data. • Define standard dimensions and designate sparse and dense storage. • Identify any need for attribute dimensions. • Identify any need for currency conversion applications that track data in different currencies. • Define calculations needed for outline dimensions and members. • Identify any need to monitor data changes in the database. You monitor data changes using the Essbase triggers feature.	
Estimate the size of your database, check disk space, and ensure that the sizes of the index, data files, and data caches in memory are adequate.	Appendix A, "Limits"
Create an application and a database.	Chapter 6, "Creating Applications and Databases"
Design a currency application.	Chapter 13, "Designing and Building Currency Conversion Applications"
Build an outline for your database.	Chapter 7, "Creating and Changing Database Outlines"
Assign alias names to your members.	Chapter 9, "Setting Dimension and Member Properties"
Build the dimensions. Decide whether your data loads will introduce new members into the outline. If so, consider dynamically building your dimensions using a rules file and a data source. If not, set up regular data loads.	• Chapter 16, "Understanding Data Loading and Dimension Building" • Chapter 17, "Working with Rules Files"
Load your data. You can load data in these ways: • Free-form • With a rules file • With Hybrid Analysis	• Chapter 16, "Understanding Data Loading and Dimension Building" • Chapter 17, "Working with Rules Files" • Chapter 18, "Using a Rules File to Perform Operations on Records, Fields, and Data" • Chapter 19, "Performing and Debugging Data Loads or Dimension Builds" • Chapter 20, "Understanding Advanced Dimension Building Concepts"
Calculate your database. • Decide on a type of calculation: outline or calculation script, or a combination • Ensure that relationships between members and member consolidations in the database outline are correct. • Consider whether tagging some members as Dynamic Calc or using Intelligent Calculation will improve calculation efficiency. • Consider which members you should tag as two-pass calculation to ensure correct calculation results.	• Chapter 21, "Calculating Essbase Databases" • Chapter 22, "Developing Formulas" • Chapter 23, "Reviewing Examples of Formulas" • Chapter 24, "Defining Calculation Order" • Chapter 26, "Dynamically Calculating Data Values" • Chapter 27, "Calculating Time Series Data" • Chapter 28, "Developing Calculation Scripts" • Chapter 29, "Reviewing Examples of Calculation Scripts"

Process Step	Reference
	● Chapter 30, “Developing Custom-Defined Calculation Macros” ● Chapter 31, “Developing Custom-Defined Calculation Functions”
Learn about dynamic calculations and how they can improve performance.	Chapter 26, “Dynamically Calculating Data Values”
View data with Spreadsheet Add-in, other Oracle tools, or third-party tools.	● See the *Oracle Essbase Spreadsheet Add-in User's Guide* ● For third-party tools, see vendor documentation
Learn about Partitioning. Think about whether your data can benefit from being decentralized into connected databases.	● Chapter 14, “Designing Partitioned Applications” ● Chapter 15, “Creating and Maintaining Partitions”
Link files or cell notes to data cells.	Chapter 11, “Linking Objects to Essbase Data”
Copy or export data subsets.	Chapter 35, “Copying Data Subsets and Exporting Data to Other Programs”
Back up and restore your data.	*Oracle Hyperion Enterprise Performance Management System Backup and Recovery Guide*
Allocate storage and specify Essbase kernel settings for your database. ● Data compression: Specify data compression on disk and the compression scheme. ● Cache sizes: You can specify the index, data file, and data cache sizes. To prevent a slowdown of the operating system, ensure that the sum of index and data cache sizes for all the active databases on the server is not more than two-thirds of the system’s RAM. ● *Cache memory locking*: You can lock the memory that is used for the index, data file, and data caches into physical memory. ● Disk volumes: You can specify the storage location of Essbase index files and data files, appropriate disk volume names, and configuration parameters. ● Isolation level: Specify committed or uncommitted access.	● Chapter 46, “Managing Database Settings” ● Chapter 47, “Allocating Storage and Compressing Data”
Generate a report. ● Choose a type of report: structured or free-form. ● Plan the elements of the report, such as page layout, column number, member identity, data value format, and title content. ● For a structured report, create page, column, and row headings (unnecessary for free-form reports). ● Create and test a report script using Administration Services Report Script Editor or any other text editor. ● Save the report on Essbase Server or on a client computer.	● Chapter 32, “Understanding Report Script Basics” ● Chapter 33, “Developing Report Scripts” ● Chapter 35, “Copying Data Subsets and Exporting Data to Other Programs”
Fine-tune your database performance and storage settings.	● Chapter 46, “Managing Database Settings” ● Chapter 47, “Allocating Storage and Compressing Data”

Process Step	Reference
	● Chapter 50, “Monitoring Performance”
Automate routine operations by using MaxL or ESSCMD.	● Chapter 49, “Using MaxL Data Definition Language” ● Appendix E, “Using ESSCMD”
Design security for your database. ● Create a security plan for your environment based on database security needs. ● Create users and groups and assign them administrative or data-access permissions, if necessary. ● Define common data access permissions at the scope of the server, applications, databases, or data-cell levels. ● To define global application or database permissions, select the relevant application or application and database and adjust the settings.	● Chapter 38, “User Management and Security” ● Chapter 39, “Controlling Access to Database Cells” ● Chapter 40, “Security Examples in Native Security Mode”
Maintain your applications.	● Chapter 43, “Running Essbase Servers, Applications, and Databases” ● Chapter 44, “Managing Applications and Databases” ● “Using Essbase Logs” on page 729 ● Chapter 46, “Managing Database Settings” ● Chapter 47, “Allocating Storage and Compressing Data” ● Chapter 48, “Ensuring Data Integrity” ● *Oracle Hyperion Enterprise Performance Management System Backup and Recovery Guide*
Analyze and improve performance and troubleshoot errors if they occur. ● Ensure that block size is not excessively large. ● Set the correct size for the index, data file, data, and calculator caches. ● Validate the database to ensure data integrity. ● Consider using partitioning to distribute data across multiple cubes for better scalability and performance. ● Ensure that disk space is adequate to allow the application to grow over time. ● Archive data from Essbase Server on a regular basis. ● Enable logging for spreadsheet update to ensure that log files are updated after archiving. ● If sorting on retrievals, increase the size of the retrieval sort buffer.	● Chapter 50, “Monitoring Performance” ● Chapter 51, “Improving Essbase Performance” ● Chapter 52, “Optimizing Essbase Caches” ● Chapter 53, “Optimizing Database Restructuring” ● Chapter 54, “Optimizing Data Loads” ● Chapter 55, “Optimizing Calculations” ● Chapter 25, “Understanding Intelligent Calculation” ● Chapter 56, “Optimizing Reports and Other Types of Retrieval”

3 Understanding Multidimensional Databases

In This Chapter

Some information in this chapter applies only to block storage databases and is not relevant to aggregate storage databases. Also see Chapter 57, “Comparison of Aggregate and Block Storage.”

OLAP and Multidimensional Databases

Online analytical processing (OLAP) is a multidimensional, multiuser, client-server computing environment for users who need to analyze enterprise data. Finance departments use OLAP for applications such as budgeting, activity-based costing (allocations), financial performance analysis, and financial modeling. Sales departments use OLAP for sales analysis and forecasting. Marketing departments use OLAP for market research analysis, sales forecasting, promotions analysis, customer analysis, and market/customer segmentation. Typical manufacturing OLAP applications include production planning and defect analysis.

Important to all of these applications is the ability to provide managers the information that they need to make effective decisions about an organization's strategic directions. A successful OLAP application provides information as needed; that is, it provides “just-in-time” information for effective decision-making.

Providing such information requires more than a base level of detailed data. Just-in-time information is computed data that usually reflects complex relationships and is often calculated on the fly. Analyzing and modeling complex relationships are practical only if response times are consistently short. In addition, because the nature of data relationships may not be known in advance, the data model must be flexible. A truly flexible data model ensures that OLAP systems can respond to changing business requirements as needed for effective decision making.

Although OLAP applications are found in widely divergent functional areas, all require the following key features:

- Multidimensional views of data
- Calculation-intensive capabilities
- Time intelligence

Key to OLAP systems are multidimensional databases, which not only consolidate and calculate data; but also provide retrieval and calculation of a variety of data subsets. A multidimensional database supports multiple views of data sets for users who need to analyze the relationships between data categories. For example, a marketing analyst might ask following questions:

- How did Product A sell last month? How does this figure compare to sales in the same month over the last five years? How did the product sell by branch, region, and territory?
- Did this product sell better in particular regions? Are there regional trends?
- Did customers return Product A last year? Were the returns due to product defects? Did the company manufacture the products in a specific plant?
- Did commissions and pricing affect how salespeople sold the product? Did certain salespeople sell more?

In multidimensional databases, the number of data views is limited only by the database outline, the structure that defines all elements of the database. Users can pivot the data to see information from a different viewpoint, drill down to find more detailed information, or drill up to see an overview.

Dimensions and Members

This section introduces the concepts of outlines, dimensions, and members within a multidimensional database. If you understand dimensions and members, you are well on your way to understanding the power of a multidimensional database.

A dimension represents the highest consolidation level in the database outline. The database outline presents dimensions and members in a tree structure to indicate a consolidation relationship. For example, in Figure 2 on page 63, Year is a dimension (of type Time) and Qtr1 is a member.

Essbase has standard dimensions and attribute dimensions.

Standard dimensions represent the core components of a business plan and often relate to departmental functions. Typical standard dimensions: Time, Accounts, Product Line, Market, and Division. Dimensions change less frequently than members.

Attribute dimensions are associated with standard dimensions. Through attribute dimensions, you group and analyze members of standard dimensions based on the member attributes (characteristics). For example, you can compare the profitability of noncaffeinated products that are packaged in glass to the profitability of noncaffeinated products packaged in cans.

Members are the individual components 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. Essbase can store the data associated with a member (referred to as a stored member in this chapter), or it can dynamically calculate the data when a user retrieves it.

Outline Hierarchies

All Essbase database development begins with creating a database outline, which accomplishes the following:

- Defines the structural relationships between members in an Essbase database
- Organizes data in the database
- Defines the consolidations and mathematical relationships between items

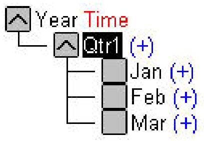
Essbase uses the concept of members to represent data hierarchies. Each dimension consists of one or more members. The members, in turn, may consist of other members. When you create a dimension, you tell Essbase how to consolidate the values of its individual members. Within the tree structure of the database outline, a consolidation is a group of members in a branch of the tree.

For example, many businesses summarize their data monthly, rolling up monthly data to obtain quarterly figures and rolling up quarterly data to obtain annual figures. Businesses may also summarize data by zip code, city, state, and country. Any dimension can be used to consolidate data for reporting purposes.

In the Sample.Basic database included with Essbase Server, for example, the Year dimension comprises five members: Qtr1, Qtr2, Qtr3, and Qtr4, each storing data for an individual quarter, plus Year, storing summary data for the year. Qtr1 comprises four members: Jan, Feb, and Mar, each storing data for a month, plus Qtr1, storing summary data for the quarter. Similarly, Qtr2, Qtr3, and Qtr4 comprise the members that represent the individual months plus the member that stores the quarterly totals.

The database outline in Figure 2 uses a hierarchical structure to represent the data consolidations and relationships in Qtr, as described in the previous paragraph.

Figure 2 Hierarchical Structure

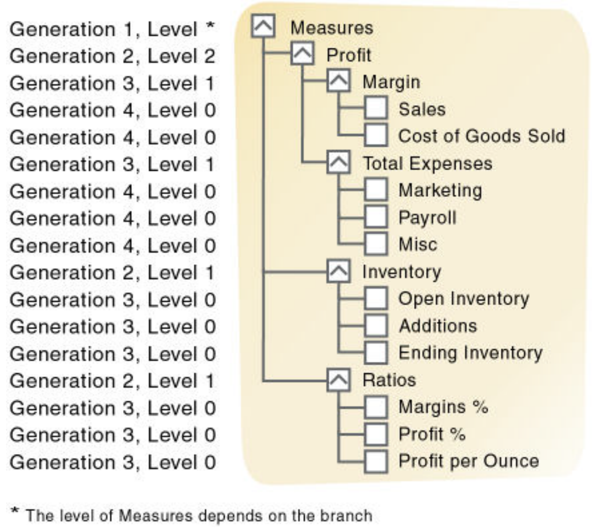


Some dimensions consist of relatively few members, while others may have hundreds or even thousands of members. Essbase does not limit the number of members within a dimension and enables the addition of new members as needed.

Dimension and Member Relationships

Essbase uses hierarchical (generations and level; and roots and leaves) and family history (parents, children, and siblings; and descendants and ancestors) terms to describe the roles and relationships of the members in a database outline. The subtopics in this section reference the outline show in Figure 3 on page 64 in describing the position of the members.

Figure 3 Member Generation and Level Numbers


Generation 1, Level * — Measures
Generation 2, Level 2 — Profit
Generation 3, Level 1 — Margin
Generation 4, Level 0 — Sales
Generation 4, Level 0 — Cost of Goods Sold
Generation 3, Level 1 — Total Expenses
Generation 4, Level 0 — Marketing
Generation 4, Level 0 — Payroll
Generation 4, Level 0 — Misc
Generation 2, Level 1 — Inventory
Generation 3, Level 0 — Open Inventory
Generation 3, Level 0 — Additions
Generation 3, Level 0 — Ending Inventory
Generation 2, Level 1 — Ratios
Generation 3, Level 0 — Margins %
Generation 3, Level 0 — Profit %
Generation 3, Level 0 — Profit per Ounce
* The level of Measures depends on the branch

Parents, Children, and Siblings

Figure 3 illustrates the following parent, child, and sibling relationships:

- A parent is a member that has a branch below it. For example, Margin is a parent member for Sales and Cost of Goods Sold.
- A child is a member that has a parent above it. For example, Sales and Cost of Goods Sold are children of the parent Margin.
- Siblings are child members of the same immediate parent, at the same generation. For example, Sales and Cost of Goods Sold are siblings (they both have the parent Margin), but Marketing (at the same branch level) is not a sibling, because its parent is Total Expenses.

Descendants and Ancestors

Figure 3 on page 64 illustrates the following descendant and ancestral relationships:

- Descendants are members in branches below a parent. For example, Profit, Inventory, and Ratios are descendants of Measures. The children of Profit, Inventory, and Ratios are also descendants of Measures.
- Ancestors are members in branches above a member. For example, Margin, Profit, and Measures are ancestors of Sales.

Roots and Leaves

Figure 3 on page 64 illustrates the following root and leaf member relationships:

- The root is the top member in a branch. Measures is the root for Profit, Inventory, Ratios, and the children of Profit, Inventory, and Ratios.
- Leaf members have no children. They are also referred to as level 0 members. For example, Opening Inventory, Additions, and Ending Inventory are leaf members.

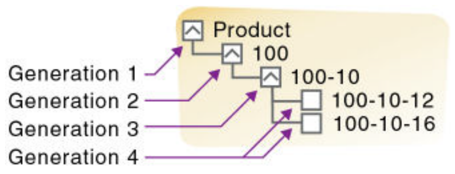
Generations and Levels

Figure 3 on page 64 illustrates the following generations levels:

- Generation refers to a consolidation level within a dimension. A root branch of the tree is generation 1. Generation numbers increase as you count from the root toward the leaf member. In Figure 3, Measures is generation 1, Profit is generation 2, and Margin is generation 3. All siblings of each level belong to the same generation; for example, both Inventory and Ratios are generation 2.

  Figure 4 shows part of the Product dimension with its generations numbered.

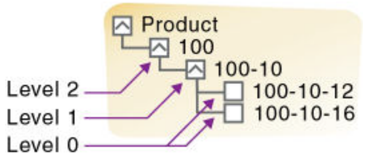
  Figure 4 Generations

- Level also refers to a branch within a dimension; levels reverse the numerical ordering used for generations. Levels count up from the leaf member toward the root. The root level number varies depending on the depth of the branch. In Figure 3, Sales and Cost of Goods Sold are level 0. All other leaf members are also level 0. Margin is level 1, and Profit is level 2. Notice that the level number of Measures varies depending on the branch. For the Ratios branch, Measures is level 2. For the Total Expenses branch, Measures is level 3.

  Figure 5 shows part of the Product dimension with its levels numbered.

  Figure 5 Levels

Generation and Level Names

To ease report maintenance, you can assign a name to a generation or level and then use the name as a shorthand for all members in that generation or level. Because changes to an outline are automatically reflected in a report, when you use generation and level names, you do not need to change the report if a member name is changed or deleted from the database outline.

Standard Dimensions and Attribute Dimensions

Essbase has standard dimensions and attribute dimensions. This chapter focuses on standard dimensions, because Essbase does not allocate storage for attribute dimension members. Instead, it dynamically calculates the members when the user requests data associated with them.

An attribute dimension is a special type of dimension that is associated with a standard dimension. See Chapter 10, "Working with Attributes."

Sparse and Dense Dimensions

Most data sets of multidimensional databases have two characteristics:

- Data is *not* smoothly and uniformly distributed.
- Data does *not* exist for the majority of member combinations. For example, all products may not be sold in all areas of the country.

Essbase maximizes performance by dividing the standard dimensions of an application into two types: dense dimensions and sparse dimensions. This division allows Essbase to cope with data that is not smoothly distributed, without losing the advantages of matrix-style access to the data. Essbase speeds data retrieval while minimizing memory and disk requirements.

Most multidimensional databases are inherently sparse; they lack data values for the majority of member combinations. A sparse dimension is one with a low percentage of available data positions filled.

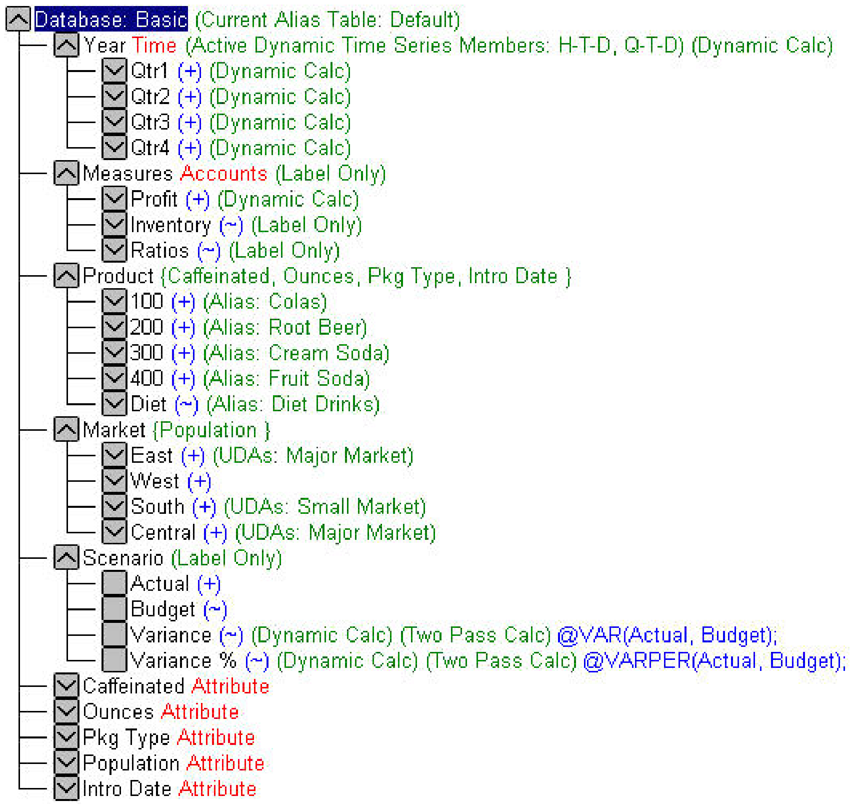
For example, the Sample.Basic database in Figure 6 includes the Year, Product, Market, Measures, and Scenario dimensions. Product represents the product units, Market represents the geographical regions in which the products are sold, and Measures represents the accounts data. Because not every product is sold in every market, Market and Product are chosen as sparse dimensions.

Most multidimensional databases also contain dense dimensions. A dense dimension has a high probability that one or more cells is occupied in every combination of dimensions. For example, in the Sample.Basic database, accounts data exists for almost all products in all markets, so Measures is chosen as a dense dimension. Year and Scenario are also chosen as dense dimensions. Year represents time in months, and Scenario represents whether the accounts values are budget or actual values.

Note:

In Figure 6, Caffeinated, Intro Date, Ounces, Pkg Type and Population are attribute dimensions. See Chapter 10, "Working with Attributes."

Figure 6 Sample.Basic Database Outline



```
Database: Basic (Current Alias Table: Default)
  Year Time (Active Dynamic Time Series Members: H-T-D, Q-T-D) (Dynamic Calc)
    Qtr1 (+) (Dynamic Calc)
    Qtr2 (+) (Dynamic Calc)
    Qtr3 (+) (Dynamic Calc)
    Qtr4 (+) (Dynamic Calc)
  Measures Accounts (Label Only)
    Profit (+) (Dynamic Calc)
    Inventory (~) (Label Only)
    Ratios (~) (Label Only)
  Product {Caffeinated, Ounces, Pkg Type, Intro Date }
    100 (+) (Alias: Colas)
    200 (+) (Alias: Root Beer)
    300 (+) (Alias: Cream Soda)
    400 (+) (Alias: Fruit Soda)
    Diet (~) (Alias: Diet Drinks)
  Market {Population }
    East (+) (UDAs: Major Market)
    West (+)
    South (+) (UDAs: Small Market)
    Central (+) (UDAs: Major Market)
  Scenario (Label Only)
    Actual (+)
    Budget (~)
    Variance (~) (Dynamic Calc) (Two Pass Calc) @VAR(Actual, Budget);
    Variance % (~) (Dynamic Calc) (Two Pass Calc) @VARPER(Actual, Budget);
  Caffeinated Attribute
  Ounces Attribute
  Pkg Type Attribute
  Population Attribute
  Intro Date Attribute
```

Selection of Dense and Sparse Dimensions

In most data sets, existing data tends to follow predictable patterns of density and sparsity. If you match patterns correctly, you can store the existing data in a reasonable number of fairly dense data blocks, rather than in many highly sparse data blocks.

By default, a new dimension is set sparse. To help you determine whether dimensions should be dense or sparse, Essbase provides an automatic configuration feature.

➤ To select automatic configuration of dense and sparse dimensions, see “Setting Dimensions as Dense or Sparse” in the *Oracle Essbase Administration Services Online Help*.

Essbase can make recommendations for the sparse-dense configuration of dimensions based on the following factors:

- The time and accounts tags on dimensions
- The probable size of the data blocks
- Characteristics that you attribute to the dimensions

You can apply a recommended configuration, or you can turn off automatic configuration and manually set the sparse or dense property for each dimension. Attribute dimensions are always sparse dimensions. Keep in mind that you can associate attribute dimensions only with sparse standard dimensions.

Note:

The automatic configuration of dense and sparse dimensions provides only an estimate. It cannot take into account the nature of the data that you will load into your database or multiple user considerations.

Dense-Sparse Configuration for Sample.Basic

Consider the Sample.Basic database, which represents data for The Beverage Company (TBC).

Because TBC does not sell every product in every market, the data set is reasonably sparse. Data values do not exist for many combinations of members in the Product and Market dimensions. For example, if Caffeine Free Cola is not sold in Florida, data values do not exist for the combination Caffeine Free Cola (100-30) -> Florida, so Product and Market are sparse dimensions. Therefore, if no data values exist for a specific combination of members in these dimensions, Essbase does not create a data block for the combination.

However, consider combinations of members in the Year, Measures, and Scenario dimensions. Data values almost always exist for some member combinations on these dimensions. For example, data values exist for the member combination Sales -> January -> Actual, because at least some products are sold in January. Thus, Year and, similarly, Measures and Scenario, are dense dimensions.

The sparse-dense configuration of the standard dimensions in the Sample.Basic database may be summarized:

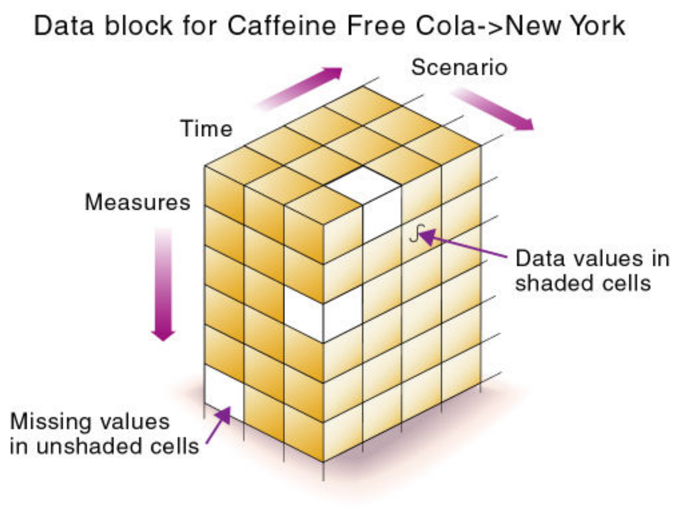
- The sparse standard dimensions are Product and Market.
- The dense standard dimensions are Year, Measures, and Scenario.

Essbase creates a data block for each unique combination of members in the Product and Market dimensions (see "Data Storage" on page 72). Each data block represents data from the dense dimensions. The data blocks likely will have few empty cells.

For example, consider the sparse member combination Caffeine Free Cola (100-30), New York, in Figure 7:

- If accounts data (represented by the Measures dimension) exists for this combination for January, it probably exists for February and for all members in the Year dimension.
- If a data value exists for one member on the Measures dimension, it is likely that other accounts data values exist for other members in the Measures dimension.
- If Actual accounts data values exist, it is likely that Budget accounts data values exist.

Figure 7 Dense Data Block for Sample.Basic Database


Data block for Caffeine Free Cola->New York
Scenario
Time
Measures
Data values in shaded cells
Missing values in unshaded cells

Dense and Sparse Selection Scenarios

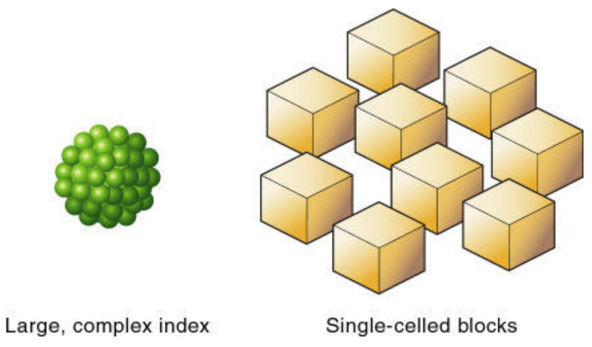
The following scenarios show how a database is affected when you select different standard dimensions. Assume that these scenarios are based on typical databases with at least seven dimensions and several hundred members:

Scenario 1: All Sparse Standard Dimensions

If you make all dimensions sparse, Essbase creates data blocks that consist of single data cells that contain single data values. An index entry is created for each data block and, therefore, in this scenario, for each existing data value.

This configuration produces an index that requires a large memory. The more index entries, the longer Essbase searches for a specific block.

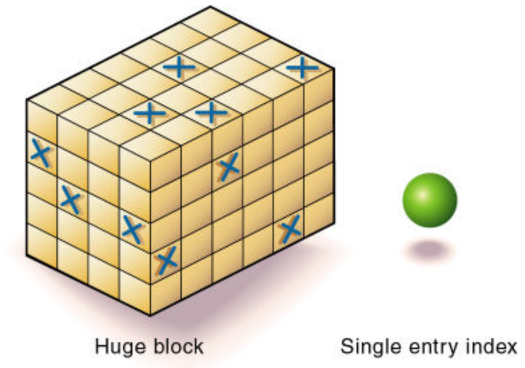
Figure 8 Database with All Sparse Standard Dimensions



Scenario 2: All Dense Standard Dimensions

If you make all dimensions dense, as shown in Figure 9, Essbase creates one index entry and one large, sparse block. In most applications, this configuration requires thousands of times more storage than other configurations. Essbase must load the entire memory when it searches for a data value, which requires enormous memory.

Figure 9 Database with All Dense Standard Dimensions

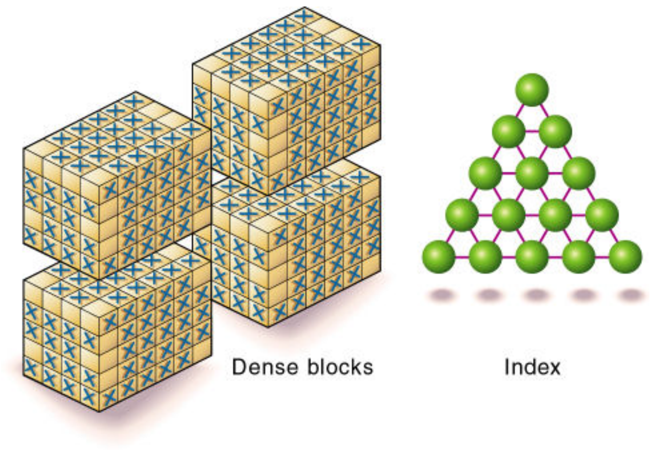


Scenario 3: Dense and Sparse Standard Dimensions

Based on your knowledge of your company's data, you have identified all your sparse and dense standard dimensions. Ideally, you have approximately equal numbers of sparse and dense standard dimensions. If not, you are probably working with a nontypical data set, and you must do more tuning to define the dimensions.

Essbase creates dense blocks that can fit into memory easily and creates a relatively small index, as shown in Figure 10. Your database runs efficiently using minimal resources.

Figure 10 An Ideal Configuration with Combination of Dense and Sparse Dimensions

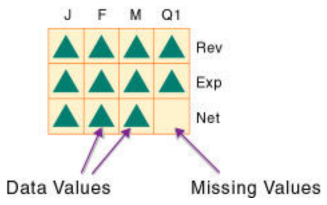


Scenario 4: A Typical Multidimensional Problem

Consider a database with four standard dimensions: Time, Accounts, Region, and Product. In the following example, Time and Accounts are dense dimensions, and Region and Product are sparse dimensions.

The two-dimensional data blocks shown in Figure 11 represent data values from the dense dimensions: Time and Accounts. The members in the Time dimension are J, F, M, and Q1. The members in the Accounts dimension are Rev, Exp, and Net.

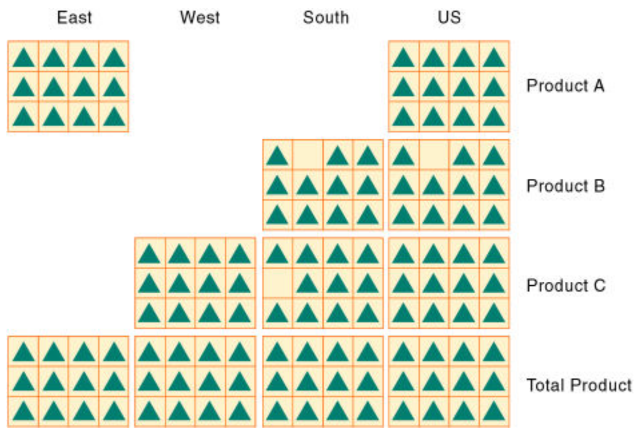
Figure 11 **Two-dimensional Data Block for Time and Accounts**



Essbase creates data blocks for combinations of members in the sparse standard dimensions (providing that at least one data value exists for the member combination). The sparse dimensions are Region and Product. The members of the Region dimension are East, West, South, and Total US. The members in the Product dimension are Product A, Product B, Product C, and Total Product.

Figure 12 shows 11 data blocks. No data values exist for Product A in the West and South, for Product B in the East and West, or for Product C in the East. Therefore, Essbase has not created data blocks for these member combinations. The data blocks that Essbase has created have few empty cells.

Figure 12 **Data Blocks Created for Sparse Members on Region and Product**

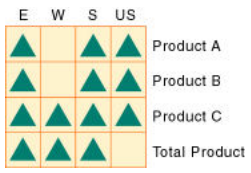


This example effectively concentrates all sparseness into the index and concentrates all data into fully utilized blocks. This configuration provides efficient data storage and retrieval.

Next, consider a reversal of the dense and sparse dimension selections. In the following example, Region and Product are dense dimensions, and Time and Accounts are sparse dimensions.

In Figure 13, the two-dimensional data blocks represent data values from the dense dimensions: Region and Product.

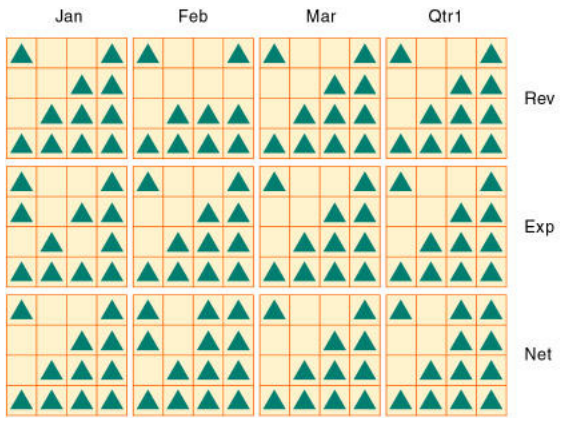
Figure 13 Two-Dimensional Data Block for Region and Product



Essbase creates data blocks for combinations of members in the sparse standard dimensions (providing that at least one data value exists for the member combination). The sparse standard dimensions are Time and Accounts.

Figure 14 shows 12 data blocks. Data values exist for all combinations of members in the Time and Accounts dimensions; therefore, Essbase creates data blocks for all member combinations. Because data values do not exist for all products in all regions, the data blocks have many empty cells. Data blocks with many empty cells store data inefficiently.

Figure 14 Data Blocks Created for Sparse Members on Time and Accounts



Data Storage

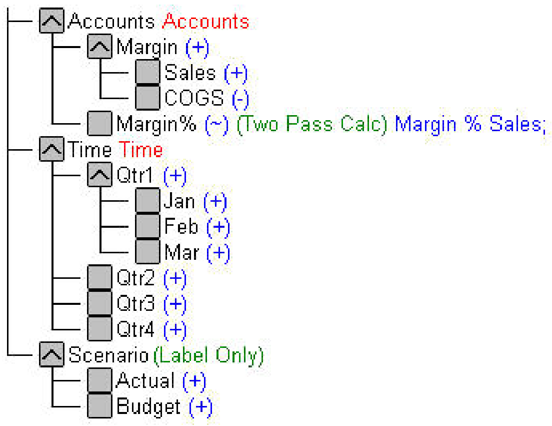
Each data value in a multidimensional database is stored in one cell. A particular data value is referenced by specifying its coordinates along *each* standard dimension.

Note:

Essbase does not store data for attribute dimensions. Essbase dynamically calculates attribute dimension data when a user retrieves the data.

Consider the simplified database shown in Figure 15.

Figure 15 A Multidimensional Database Outline



```
Accounts Accounts
    Margin (+)
        Sales (+)
        COGS (-)
    Margin% (~) (Two Pass Calc) Margin % Sales;
Time Time
    Qtr1 (+)
        Jan (+)
        Feb (+)
        Mar (+)
    Qtr2 (+)
    Qtr3 (+)
    Qtr4 (+)
Scenario (Label Only)
    Actual (+)
    Budget (+)
```

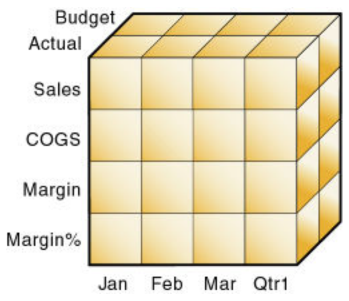
This database has three dimensions: Accounts, Time, and Scenario:

- The Accounts dimension has four members: Sales, COGS, Margin, and Margin%.
- The Time dimension has four quarter members, and Qtr1 has three month members
- The Scenario dimension has two child members: Budget for budget values and Actual for actual values.

Data Values

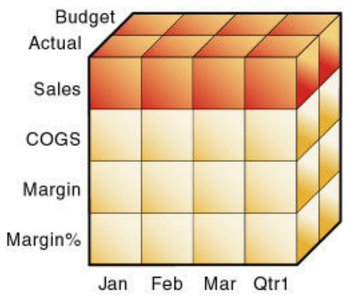
The intersection of one member from one dimension with one member from each of the other dimensions represents a data value. The example in Figure 16 has three dimensions; therefore, the dimensions and data values in the database can be represented in a cube.

Figure 16 Three-Dimensional Database



The shaded cells in Figure 17 illustrate that when you specify Sales, you are specifying the portion of the database containing eight Sales values.

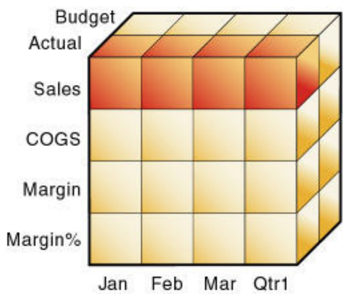
Figure 17 Sales Slice of the Database



Slicing a database amounts to fixing one or more dimensions at a constant value while allowing the other dimensions to vary.

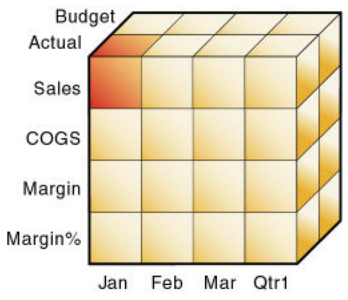
When you specify Actual Sales, you are specifying the four Sales values where Actual and Sales intersect, as shown by the shaded area in Figure 18.

Figure 18 Actual, Sales Slice of the Database



A data value is stored in one cell in the database. To refer to a specific data value in a multidimensional database, you specify its member on each dimension. In Figure 19, the cell containing the data value for Sales, Jan, Actual is shaded. The data value can also be expressed using the cross-dimensional operator (->) as Sales -> Actual -> Jan.

Figure 19 Sales -> Jan -> Actual Slice of the Database



Data Blocks and the Index System

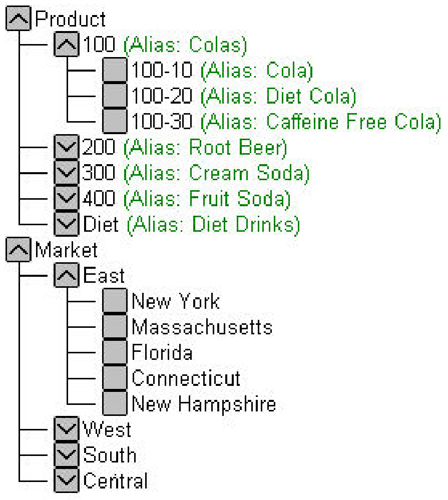
Essbase uses two types of internal structures to store and access data: data blocks and the index system.

Essbase creates a data block for each unique combination of sparse standard dimension members (providing that at least one data value exists for the sparse dimension member combination). The data block represents all the dense dimension members for its combination of sparse dimension members.

Essbase creates an index entry for each data block. The index represents the combinations of sparse standard dimension members. It contains an entry for each unique combination of sparse standard dimension members for which at least one data value exists.

For example, in the Sample.Basic database outline shown in Figure 20, Product and Market are sparse dimensions.

Figure 20 Product and Market Dimensions from the Sample.Basic Database



```
Product
  100 (Alias: Colas)
    100-10 (Alias: Cola)
    100-20 (Alias: Diet Cola)
    100-30 (Alias: Caffeine Free Cola)
  200 (Alias: Root Beer)
  300 (Alias: Cream Soda)
  400 (Alias: Fruit Soda)
  Diet (Alias: Diet Drinks)
Market
  East
    New York
    Massachusetts
    Florida
    Connecticut
    New Hampshire
  West
  South
  Central
```

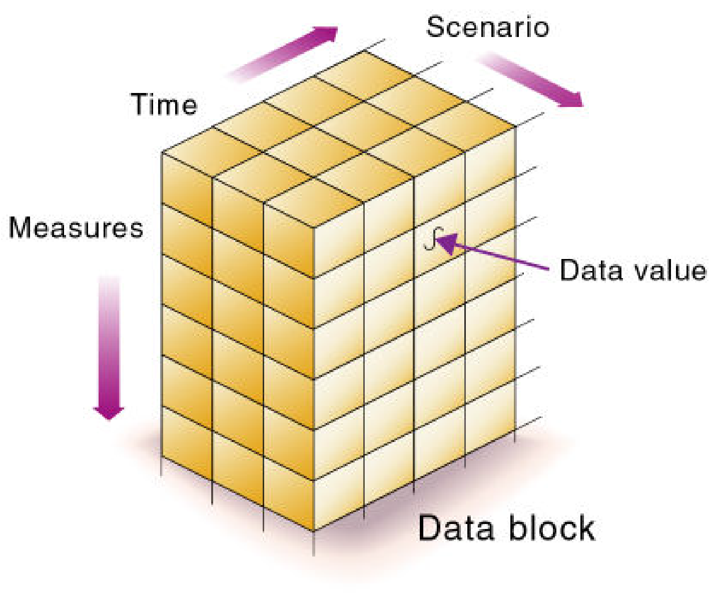
If data exists for Caffeine Free Cola in New York, Essbase creates a data block and an index entry for the sparse member combination of Caffeine Free Cola (100-30) -> New York. If Caffeine Free Cola is *not* sold in Florida, Essbase does *not* create a data block or an index entry for the sparse member combination: Caffeine Free Cola (100-30) -> Florida.

The data block Caffeine Free Cola (100-30) -> New York represents all the Year, Measures, and Scenario dimensions for Caffeine Free Cola (100-30) -> New York.

Each unique data value can be considered to exist in a cell in a data block. When Essbase searches for a data value, it uses the index to locate the appropriate data block. Then, within the data block, it locates the cell containing the data value. The index entry provides a pointer to the data block. The index handles sparse data efficiently because it includes only pointers to existing data blocks.

Figure 21 shows part of a data block for the Sample.Basic database. Each dimension of the block represents a dense dimension in the Sample.Basic database: Time, Measures, and Scenario. A data block exists for each unique combination of members of the Product and Market sparse dimensions (providing that at least one data value exists for the combination).

Figure 21 Part of a Data Block for the Sample.Basic Database



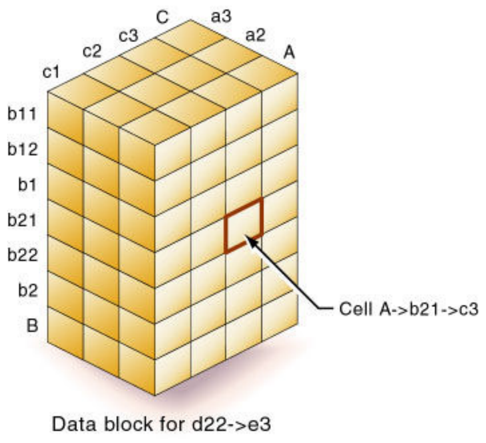
Each data block is a multidimensional array that contains a fixed, ordered location for each possible combination of dense dimension members. Accessing a cell in the block does not involve sequential or index searches. The search is almost instantaneous, resulting in optimal retrieval and calculation speed.

Essbase orders the cells in a data block according to the order of the members in the dense dimensions of the database outline.

```
A (Dense)
   a1
   a2
B (Dense)
   b1
       b11
       b12
   b2
       b21
       b22
C (Dense)
   c1
   c2
   c3
D (Sparse)
   d1
   d2
       d21
       d22
E (Sparse)
   e1
   e2
   e3
```

The block in Figure 22 represents the three dense dimensions from within the combination of the sparse members d22 and e3 in the preceding database outline. In Essbase, member combinations are denoted by the cross-dimensional operator. The symbol for the cross-dimensional operator is ->, so d22 -> e3 denotes the block for d22 and e3. The intersection of A, b21, and c3 is written A -> b21 -> c3.

Figure 22 Data Block Representing Dense Dimensions for d22 -> e3



Essbase creates a data block for every unique combination of the members of the sparse dimensions D and E (providing that at least one data value exists for the combination).

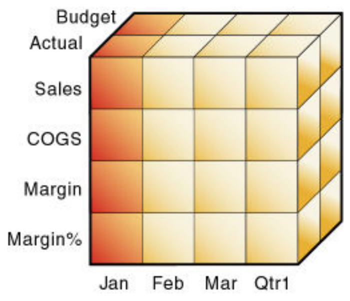
Data blocks, such as the one in Figure 22, may include cells that do not contain data values. A data block is created if at least one data value exists in the block. Essbase compresses data blocks with missing values on disk, expanding each block fully as it brings the block into memory. Data compression is optional but is enabled by default. See "Data Compression" on page 772.

By carefully selecting dense and sparse standard dimensions, you can ensure that data blocks do not contain many empty cells, minimizing disk storage requirements and improving performance. In Essbase, empty cells are known as #MISSING data.

Multiple Data Views

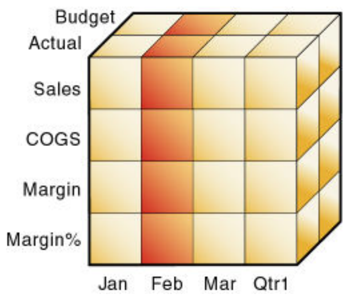
A multidimensional database supports multiple views of data sets for users who need to analyze the relationships between data categories. Slicing the database in different ways gives you different perspectives of the data. The slice of January in Figure 23, for example, examines all data values for which the Year dimension is fixed at Jan.

Figure 23 Data for January



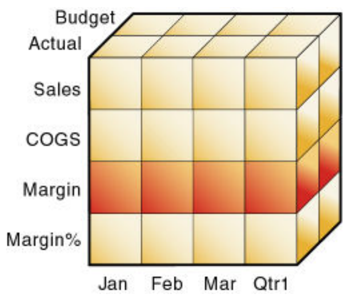
The slice in Figure 24 shows data for the month of February:

Figure 24 Data for February



The slice in Figure 25 shows data for profit margin:

Figure 25 Data for Profit Margin



The Essbase Solution

To create an optimized Essbase database, ask:

- How does your company use the data?
- How will you build and order the dimensions?
- Which data compression scheme will you use?
- How will you create and order calculations?

See these topics:

- Planning the development of your multidimensional database, see Chapter 4, "Case Study: Designing a Single-Server, Multidimensional Database."
- Selecting dense and sparse dimensions, see "Selection of Dense and Sparse Dimensions" on page 67.
- Loading data, see Chapter 16, "Understanding Data Loading and Dimension Building."
- Compressing data and optimizing your database, see "Data Compression" on page 772.
- Calculating your database, see Chapter 21, "Calculating Essbase Databases."

4 Case Study: Designing a Single-Server, Multidimensional Database

In This Chapter

Some information in this chapter applies only to block storage databases and is not relevant to aggregate storage databases. Also see Chapter 57, “Comparison of Aggregate and Block Storage.”

Process for Designing a Database

To implement a multidimensional database, you install Essbase, and then you design and create an application and databases. You analyze data sources and define requirements carefully and decide whether a single-server approach or a partitioned, distributed approach better serves your needs. For criteria that you can review to decide whether to partition an application, see “Guidelines for Partitioning a Database” on page 217.

Using a case study, this chapter provides an overview of the database planning process and discusses working rules that you can follow to design a single-server, multidimensional database solution for your organization. See Chapter 6, “Creating Applications and Databases.”

Figure 26 on page 80 illustrates the cyclical process of designing a database, which includes the following basic steps:

1. Analyze business needs and design a plan.

   The application and database that you create must satisfy the information needs of your users and your organization. Therefore, you identify source data, define user information access needs, review security considerations, and design a database model. See “Analyzing and Planning” on page 81.

2. Draft a database outline.

The outline determines the structure of the database—what information is stored and how different pieces of information interrelate. See "Drafting Outlines" on page 92.

3. Check system requirements.

   How you meet system requirements and define system parameters affects the efficiency and performance of the database. See "Checking System Requirements" on page 96.

4. Load test data into the database.

   After an outline and a security plan are in place, you load the database with test data to enable the later steps of the process. See "Loading Test Data" on page 97.

5. Define calculations.

   You test outline consolidations and write and test formulas and calculation scripts for specialized calculations. See "Defining Calculations" on page 97.

6. Define reports.

   Users access data through print and online reports and spreadsheets or on the Web. If you plan to provide predefined reports to users, you design report layouts and run reports. See "Defining Reports" on page 106.

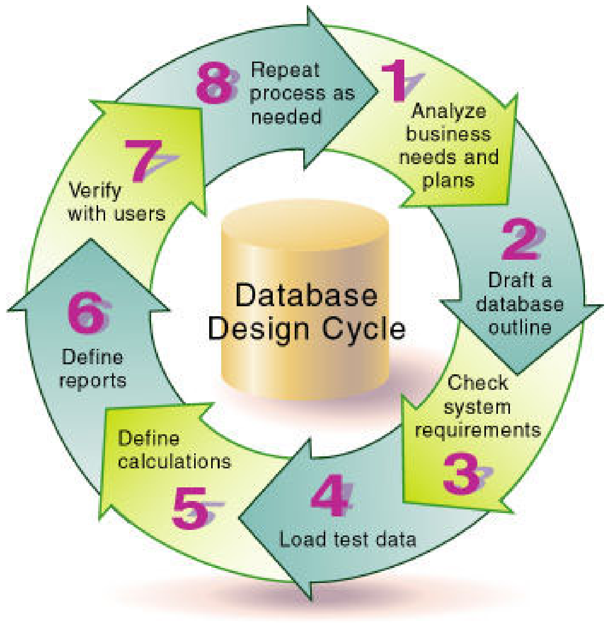
7. Verify with users.

   To ensure that the database satisfies your user goals, solicit and carefully consider user opinions. See "Verifying the Design" on page 106.

8. Repeat the process.

   To fine-tune the design, repeat steps 1 through 7.

Figure 26 The Database Design Cycle


Database Design Cycle
1 Analyze business needs and plans
2 Draft a database outline
Check system requirements 3
4 Load test data
Define calculations 5
6 Define reports
7 Verify with users
8 Repeat process as needed

Case Study: The Beverage Company

This chapter bases the database planning process on the needs of a fictitious company called *The Beverage Company* (TBC) and uses TBC as an example to demonstrate how to build an Essbase database. TBC is a variation of the Sample.Basic application that is included with the Essbase installation.

TBC manufactures, markets, and distributes soft drink products internationally. Analysts at TBC prepare budget forecasts and compare performance to budget forecasts monthly. The financial measures that analysts track are profit and loss and inventory.

TBC uses spreadsheet packages to prepare budget data and perform variance reporting. Because TBC plans and tracks a variety of products over several markets, the process of deriving and analyzing data is tedious. Last month, analysts spent most of their time entering and rekeying data and preparing reports.

TBC has determined that Essbase is the best tool for creating a centralized repository for financial data. The data repository will reside on a server that is accessible to analysts throughout the organization. Users can access the server and load data from various sources and retrieve data as needed. TBC has a variety of users, so TBC expects that different users will have different security levels for accessing data.

Analyzing and Planning

The design and operation of an Essbase multidimensional database are key to achieving a well-tuned system that enables you to analyze business information efficiently. Given the size and performance volatility of multidimensional databases, developing an optimized database is critical. A detailed plan that outlines data sources, user needs, and prospective database elements can save you development and implementation time.

The planning and analysis phase involves these tasks:

- "Analyzing Source Data" on page 82
- "Identifying User Requirements" on page 82
- "Planning for Security in a Multiple User Environment" on page 83
- "Creating Database Models" on page 83

When designing a multidimensional application, consider these factors:

- How information flows within the company—who uses which data for what purposes
- The types of reporting the company does—what types of data must be included in the outline to serve user reporting needs

Note:

Defining only one database per application enables enhanced memory usage and ease of database administration. Applications that use the optional Essbase Currency Conversion module are an exception to this recommendation. Currency conversion applications generally consist of a main database and a separate currency database (see Chapter 13, "Designing and Building Currency Conversion Applications").

Analyzing Source Data

First, evaluate the source data to be included in the database. Think about where the data resides and how often you plan to update the database with the data. This up-front research saves time when you create the database outline and load data into the Essbase database.

Determine the scope of the database. If an organization has thousands of product families containing hundreds of thousands of products, you may want to store data values only for product families. Interview members from each user department to find out what data they process, how they process data today, and how they want to process data in the future.

Carefully define reporting and analysis needs.

- How do users want to view and analyze data?
- How much detail should the database contain?
- Does the data support the desired analysis and reporting goals?
- If not, what additional data do you need, and where can you find it?

Determine the location of the current data.

- Where does each department currently store data?
- Is data in a form that Essbase can use?
- Do departments store data in relational databases on Windows or UNIX servers, or in Excel spreadsheets?
- Who updates the database and how frequently?
- Do those who need to update data have access to it?

Ensure that the data is ready to load into Essbase.

- Does data come from a single source or multiple sources?
- Is data in a format that Essbase can use? For a list of valid data sources that you can load into Essbase, see "Data Sources" on page 258.
- Is all data that you want to use readily available?

Identifying User Requirements

Discuss information needs with users. Review the information they use and the reports they must generate for review by others. Determine the following requirements:

- What types of analysis do users require?
- What summary and detail levels of information do users need?
- Do some users require access to information that other users should not see?

Planning for Security in a Multiple User Environment

Consider user information needs when you plan how to set up security permissions. End your analysis with a list of users and permissions.

Use this checklist to plan for security:

- Who are the users and what permissions should they have?
- Who should have load data permissions?
- Which users can be grouped, and as a group, given similar permissions?

See Chapter 38, "User Management and Security".

Creating Database Models

Next, create a model of the database on paper. To build the model, identify the perspectives and views that are important to your business. These views translate into the dimensions of the database model.

Most businesses analyze the following areas:

- Time periods
- Accounting measures
- Scenarios
- Products
- Distribution channels
- Geographical regions
- Business units

Use the following topics to help you gather information and make decisions.

Identifying Analysis Objectives

After you identify the major areas of information in a business, the next step in designing an Essbase database is deciding how the database enables data analysis:

- If analyzing by time, which time periods are needed? Should the analysis include only the current year or multiple years? Should the analysis include quarterly and monthly data? Should it include data by season?
- If analyzing by geographical region, how do you define the regions? Do you define regions by sales territories? Do you define regions by geographical boundaries such as states and cities?
- If analyzing by product line, should you review data for each product? Can you summarize data into product classes?

Regardless of the business area, you must determine the perspective and detail needed in the analysis. Each business area that you analyze provides a different view of the data.

Determining Dimensions and Members

You can represent each business view as a separate standard dimension in the database. If you need to analyze a business area by classification or attribute, such as by the size or color of products, you can use attribute dimensions to represent the classification views.

The dimensions that you choose determine what types of analysis you can perform on the data. With Essbase, you can use as many dimensions as you need for analysis. A typical Essbase database contains at least seven standard dimensions (nonattribute dimensions) and many more attribute dimensions.

When you know approximately what dimensions and members you need, review the following topics and develop a tentative database design:

- "Relationships Among Dimensions" on page 84
- "Example Dimension-Member Structure" on page 85
- "Checklist for Determining Dimensions and Members" on page 86

After you determine the dimensions of the database model, choose the elements or items within the perspective of each dimension. These elements become the members of their respective dimensions. For example, a perspective of time may include the time periods that you want to analyze, such as quarters, and within quarters, months. Each quarter and month becomes a member of the dimension that you create for time. Quarters and months represent a two-level hierarchy of members and their children. Months within a quarter consolidate to a total for each quarter.

Relationships Among Dimensions

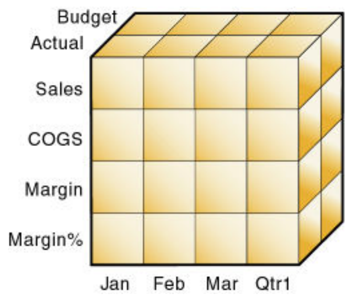
Next, consider the relationships among the business areas. The structure of an Essbase database makes it easy for users to analyze information from many perspectives. A financial analyst, for example, may ask the following questions:

- What are sales for a particular month? How does this figure compare to sales in the same month over the last five years?
- By what percentage is profit margin increasing?
- How close are actual values to budgeted values?

In other words, the analyst may want to examine information from three perspectives—time, account, and scenario. The sample database in Figure 27 represents these three perspectives as three dimensions, with one dimension represented along each of the three axes:

- A time dimension—which comprises Jan, Feb, Mar, and the total for Qtr1—is displayed along the X-axis.
- An accounts dimension, which consists of accounting figures such as Sales, COGS, Margin, and Margin%, is displayed along the Y-axis.
- Another dimension, which provides a different point of view, such as Budget for budget values and Actual for actual values, is displayed along the Z-axis.

Figure 27 Cube Representing Three Database Dimensions



The cells within the cube, where the members intersect, contain the data relevant to all three intersecting members; for example, the actual sales in January.

Example Dimension-Member Structure

Table 2 shows a summary of the TBC business areas that the planner determined would be dimensions. The dimensions represent the major business areas to be analyzed. The planner created three columns, with the dimensions in the left column and members in the two right columns. The members in column 3 are subcategories of the members in column 2. In some cases, members in column 3 are divided into another level of subcategories; for example, the Margin of the Measures dimension is divided into Sales and COGS.

Table 2 TBC Sample Dimensions

Dimensions	Members	Child Members
Year	Qtr1	Jan, Feb, Mar
	Qtr2	Apr, May, Jun
	Qtr3	Jul, Aug, Sep
	Qtr4	Oct, Nov, Dec
Measures	Profit	Margin: Sales, COGS Total Expenses: Marketing, Payroll, Miscellaneous
	Inventory	Opening Inventory, Additions, Ending Inventory
	Ratios	Margin %, Profit %, Profit per Ounce
Product	Colas (100)	Cola (100-10), Diet Cola (100-20), Caffeine Free Cola (100-30)
	Root Beer (200)	Old Fashioned (200-10), Diet Root Beer (200-20), Sarsaparilla (200-30), Birch Beer (200-40)
	Cream Soda (300)	Dark Cream (300-10), Vanilla Cream (300-20), Diet Cream Soda (300-30)
	Fruit Soda (400)	Grape (400-10), Orange (400-20), Strawberry (400-30)
Market	East	Connecticut, Florida, Massachusetts, New Hampshire, New York
	West	California, Nevada, Oregon, Utah, Washington

Dimensions	Members	Child Members
	South	Louisiana, New Mexico, Oklahoma, Texas
	Central	Colorado, Illinois, Iowa, Missouri, Ohio, Wisconsin
Scenario	Actual	
	Budget	
	Variance	
	Variance %	

In addition, the planner added two attribute dimensions to enable product analysis based on size and packaging:

Table 3 TBC Sample Attribute Dimensions

Dimensions	Members	Child Members
Ounces	Large Small	64, 32, 20 16, 12
Pkg Type	Bottle Can	

Checklist for Determining Dimensions and Members

Use the following checklist when determining the dimensions and members of your model database:

- What are the candidates for dimensions?
- Do any of the dimensions classify or describe other dimensions? These dimensions are candidates for attribute dimensions.
- Do users want to qualify their view of a dimension? The categories by which they qualify a dimension are candidates for attribute dimensions.
- What are the candidates for members?
- How many levels does the data require?
- How does the data consolidate?

Analyzing Database Design

While the initial dimension design is still on paper, you should review the design according to a set of guidelines. The guidelines help you fine-tune the database and leverage the multidimensional technology. The guidelines are processes or questions that help you achieve an efficient design and meet consolidation and calculation goals.

The number of members needed to describe a potential data point should determine the number of dimensions. If you are not sure whether you should delete a dimension, keep it and apply more analysis rules until you feel confident about deleting or keeping it.

Use the information in the following topics to analyze and improve your database design.

Dense and Sparse Dimensions

Which dimensions are sparse and which dense affects performance. For an introduction, see "Sparse and Dense Dimensions" on page 66. See "Designing an Outline to Optimize Performance" on page 95.

Standard and Attribute Dimensions

For simplicity, the examples in this topic show alternative arrangements for what initially was designed as two dimensions. You can apply the same logic to all combinations of dimensions.

Consider the design for a company that sells products to multiple customers over multiple markets; the markets are unique to each customer:

```
             Cust A  Cust B  Cust C
New York     100     N/A     N/A
Illinois     N/A     150     N/A
California   N/A     N/A     30
```

Cust A is only in New York, Cust B is only in Illinois, and Cust C is only in California. The company can define the data in one standard dimension:

```
Market
       New York
             Cust A
       Illinois
             Cust B
       California
             Cust C
```

However, if you look at a larger sampling of data, you may see that many customers can be in each market. Cust A and Cust E are in New York; Cust B, Cust M, and Cust P are in Illinois; Cust C and Cust F are in California. In this situation, the company typically defines the large dimension, Customer, as a standard dimension and the smaller dimension, Market, as an attribute dimension. The company associates the members of the Market dimension as attributes of the members of the Customer dimension. The members of the Market dimension describe locations of the customers.

```
Customer (Standard dimension)
       Cust A   (Attribute:New York)
       Cust B   (Attribute:Illinois)
       Cust C   (Attribute:California)
       Cust E   (Attribute:New York)
       Cust F   (Attribute:California)
       Cust M   (Attribute:Illinois)
       Cust P   (Attribute:Illinois)
Market (Attribute dimension)
       New York
       Illinois
       California
```

Consider another situation. Again, the company sells products to multiple customers over multiple markets, but the company can ship to a customer that has locations in different markets:

```
             Cust A   Cust B   Cust C

New York     100       75      N/A
Illinois     N/A      150      N/A
California   150      N/A       30
```

Cust A is in New York and California. Cust B is in New York and Illinois. Cust C is only in California. Using an attribute dimension does not work in this situation; a customer member cannot have more than one attribute member. Therefore, the company designs the data in two standard dimensions:

```
Customer
       Cust A
       Cust B
       Cust C
Market
       New York
       Illinois
       California
```

Dimension Combinations

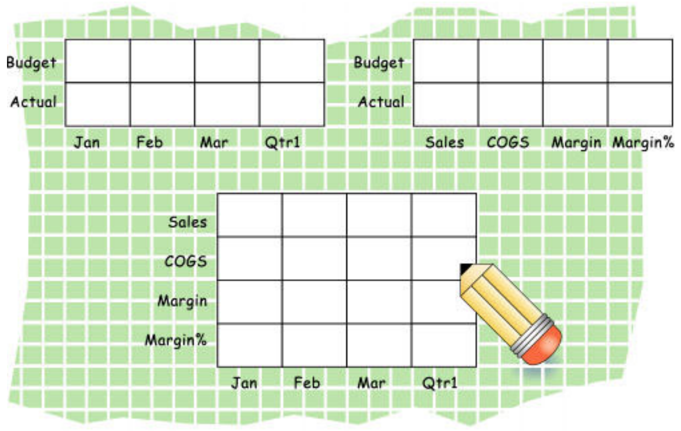
Break each combination of two dimensions into a two-dimensional matrix. For example, proposed dimensions at TBC (as listed in Table 2 on page 85) include the following combinations:

- Year across Measures
- Year across Product
- Year across Market
- Year across Scenario
- Measures across Product
- Measures across Market
- Measures across Scenario
- Market across Product
- Market across Scenario
- Scenario across Product
- Ounces across Pkg Type

Ounces and Pkg Type, as attribute dimensions associated with the Product dimension, can be considered with the Product dimension.

To help visualize each dimension, draw a matrix and include a few of the first-generation members. Figure 28 shows a simplified set of matrixes for three dimensions.

Figure 28 Analyzing Dimensional Relationships



For each combination of dimensions, ask three questions:

- Does it add analytic value?
- Does it add utility for reporting?
- Does it avoid an excess of unused combinations?

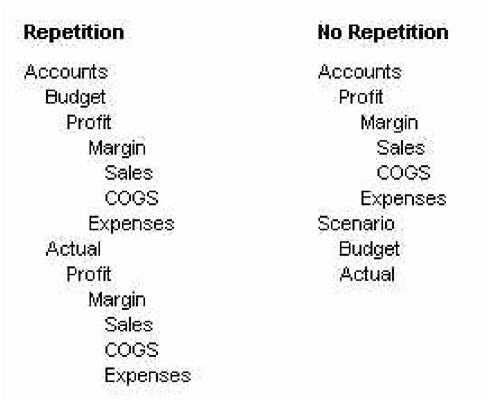
For each combination, the answers to the questions help determine whether the combination is valid for the database. Ideally, the answer to each question is yes. If not, consider rearranging the data into more-meaningful dimensions. As you work through this process, discuss information needs with users.

Repetition in Outlines

The repetition of elements in an outline often indicates a need to split dimensions. The following examples show you how to avoid repetition.

In Figure 29 on page 90, the left column, labeled “Repetition,” shows Profit, Margin, Sales, COGS, and Expenses repeated under Budget and Actual in the Accounts dimension. The right column, labeled “No Repetition,” separates Budget and Actual into another dimension (Scenario), leaving just one set of Profit, Margin, Sales, COGS, and Expenses members in the Accounts dimension. This approach simplifies the outline and provides a simpler view of the budget and actual figures of the other dimensions in the database.

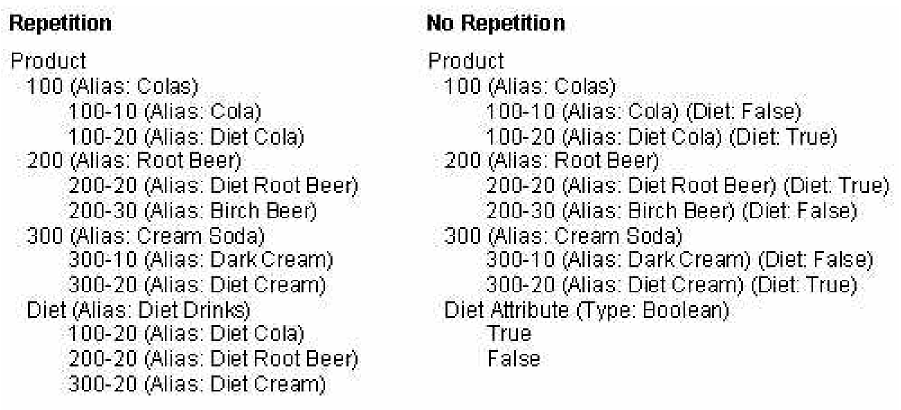
Figure 29 Example of Eliminating Repetition By Creating a Scenario Dimension



```
Repetition                     No Repetition

Accounts                       Accounts
  Budget                         Profit
    Profit                         Margin
      Margin                         Sales
        Sales                        COGS
        COGS                       Expenses
      Expenses                 Scenario
  Actual                         Budget
    Profit                       Actual
      Margin
        Sales
        COGS
        Expenses
```

In Figure 30 on page 90, the left column, labeled "Repetition," uses shared members in the Diet dimension to analyze diet beverages. Members 100–20, 200–20, and 300–20 are repeated: once under Diet, and once under their respective parents. The right column, labeled "No Repetition," simplifies the outline by creating a Diet attribute dimension of type Boolean (True or False). All members are shown only once, under their respective parents, and are tagged with the appropriate attribute ("Diet: True" or "Diet: False").

Figure 30 Example of Eliminating Repetition By Creating an Attribute Dimension



```
Repetition                              No Repetition

Product                                 Product
  100 (Alias: Colas)                      100 (Alias: Colas)
      100-10 (Alias: Cola)                    100-10 (Alias: Cola) (Diet: False)
      100-20 (Alias: Diet Cola)               100-20 (Alias: Diet Cola) (Diet: True)
  200 (Alias: Root Beer)                  200 (Alias: Root Beer)
      200-20 (Alias: Diet Root Beer)          200-20 (Alias: Diet Root Beer) (Diet: True)
      200-30 (Alias: Birch Beer)              200-30 (Alias: Birch Beer) (Diet: False)
  300 (Alias: Cream Soda)                 300 (Alias: Cream Soda)
      300-10 (Alias: Dark Cream)              300-10 (Alias: Dark Cream) (Diet: False)
      300-20 (Alias: Diet Cream)              300-20 (Alias: Diet Cream) (Diet: True)
  Diet (Alias: Diet Drinks)               Diet Attribute (Type: Boolean)
      100-20 (Alias: Diet Cola)               True
      200-20 (Alias: Diet Root Beer)          False
      300-20 (Alias: Diet Cream)
```

Attribute dimensions also provide additional analytic capabilities. See "Designing Attribute Dimensions" on page 168.

Interdimensional Irrelevance

Interdimensional irrelevance occurs when many members of a dimension are irrelevant across other dimensions. Essbase defines irrelevant data as data that Essbase stores only at the summary (dimension) level. In such a situation, you may be able to remove a dimension from the database and add its members to another dimension or split the model into separate databases.

For example, TBC considered analyzing salaries as a member of the Measures dimension. But salary information often proves irrelevant in the context of a corporate database. Most salaries are confidential and apply to individuals. The individual and the salary typically represent one cell, with no reason to intersect with any other dimension.

TBC considered separating employees into a separate dimension. Table 4 shows an example of how TBC analyzed the proposed Employee dimension for interdimensional irrelevance. Members of the proposed Employee dimension (represented in the table header row) are compared with members of the Measures dimension (represented in the left-most column). An "X" in a cell indicates relevance. Only the Salary measure is relevant to individual employees.

Table 4 Example of Interdimensional Irrelevance

	Joe Smith	Mary Jones	Mike Garcia	All Employees
Revenue				x
Variable Costs				x
COGS				x
Advertising				x
Salaries	x	x	x	x
Fixed Costs				x
Expenses				x
Profit				x

Reasons to Split Databases

Because individual employee information is irrelevant to the other information in the database, and also because adding an Employee dimension would substantially increase database storage needs, TBC created a separate Human Resources (HR) database. The new HR database contains a group of related dimensions and includes salaries, benefits, insurance, and 401(k) plans.

There are many reasons for splitting a database; for example, suppose that a company maintains an organizational database that contains several international subsidiaries in several time zones. Each subsidiary relies on time-sensitive financial calculations. You can split the database for groups of subsidiaries in the same time zone to ensure that financial calculations are timely. You can also use a partitioned application to separate information by subsidiary.

Checklist to Analyze the Database Design

Use the following checklist to analyze the database design:

- Have you minimized the number of dimensions?
- For each dimensional combination, did you ask:
  - Does it add analytic value?
  - Does it add utility for reporting?
  - Does it avoid an excess of unused combinations?
- Did you avoid repetition in the outline?

- Did you avoid interdimensional irrelevance?
- Did you split the databases as necessary?

Drafting Outlines

Now you can create the application and database and build the first draft of the outline in Essbase. The draft defines all dimensions, members, and consolidations. Use the outline to design consolidation requirements and identify where you need formulas and calculation scripts.

Note:

Before you create a database and build its outline, create an Essbase application to contain it.

The TBC planners issued the following draft for a database outline. In this plan, the bold words are the dimensions—Year, Measures, Product, Market, Scenario, Pkg Type, and Ounces. Observe how TBC anticipated consolidations, calculations and formulas, and reporting requirements. The planners also used product codes rather than product names to describe products.

- **Year.** TBC needs to collect data monthly and summarize the monthly data by quarter and year. Monthly data, stored in members such as Jan, Feb, and Mar, consolidates to quarters. Quarterly data, stored in members such as Qtr1 and Qtr2, consolidates to Year.
- **Measures.** Sales, Cost of Goods Sold, Marketing, Payroll, Miscellaneous, Opening Inventory, Additions, and Ending Inventory are standard measures. Essbase can calculate Margin, Total Expenses, Profit, Total Inventory, Profit %, Margin %, and Profit per Ounce from these measures. TBC needs to calculate Measures on a monthly, quarterly, and yearly basis.
- **Product.** The Product codes are 100-10, 100-20, 100-30, 200-10, 200-20, 200-30, 200-40, 300-10, 300-20, 300-30, 400-10, 400-20, and 400-30. Each product consolidates to its respective family (100, 200, 300, and 400). Each consolidation allows TBC to analyze by size and package, because each product is associated with members of the Ounces and Pkg Type attribute dimensions.
- **Market.** Several states make up a region; four regions make up a market. The states are Connecticut, Florida, Massachusetts, New Hampshire, New York, California, Nevada, Oregon, Utah, Washington, Louisiana, New Mexico, Oklahoma, Texas, Colorado, Illinois, Iowa, Missouri, Ohio, and Wisconsin. Each state consolidates into its region—East, West, South, or Central. Each region consolidates into Market.
- **Scenario.** TBC derives and tracks budget versus actual data. Managers must monitor and track budgets and actuals, as well as the variance and variance percentage between them.
- **Pkg Type.** TBC wants to see the effect that product packaging has on sales and profit. Establishing the Pkg Type attribute dimension enables users to analyze product information based on whether a product is packaged in bottles or cans.
- **Ounces.** TBC sells products in different sizes in ounces in different markets. Establishing the Ounces attribute dimension helps users monitor which sizes sell better in which markets.

The following topics present a review of the basics of dimension and member properties and a discussion of how outline design affects performance.

Dimension and Member Properties

The properties of dimensions and members define the roles of the dimensions and members in the design of the multidimensional structure. These properties include the following:

- Dimension types and attribute associations. See "Dimension Types" on page 93.
- Data storage properties. See "Member Storage Properties" on page 94.
- Consolidation operators. See "Consolidation of Dimensions and Members" on page 97.
- Formulas. See "Formulas and Functions" on page 102.

For a complete list of dimension and member properties, see Chapter 9, "Setting Dimension and Member Properties."

Dimension Types

A dimension type is a property that Essbase provides that adds special functionality to a dimension. The most commonly used dimension types: time, accounts, and attribute. This topic uses the following dimensions of the TBC database to illustrate dimension types.

```
Database:Design
   Year (Type: time)
   Measures (Type: accounts)
   Product
   Market
   Scenario
   Pkg Type (Type: attribute)
   Ounces (Type: attribute)
```

Table 5 defines each Essbase dimension type.

Table 5 Dimension Types

Dimension Types	Description
None	Specifies no particular dimension type.
Time	Defines the time periods for which you report and update data. You can tag only one dimension as time. The time dimension enables several accounts dimension functions, such as first and last time balances.
Accounts	Contains items that you want to measure, such as profit and inventory, and makes Essbase built-in accounting functionality available. Only one dimension can be defined as accounts. For discussion of two forms of account dimension calculation, see "Accounts Dimension Calculations" on page 100.
Attribute	Contains members that can be used to classify members of another, associated dimension.

Dimension Types	Description
	For example, the Pkg Type attribute dimension contains a member for each type of packaging, such as bottle or can, that applies to members of the Product dimension.
Country	Contains data about where business activities take place. In a country dimension, you can specify the currency used in each member. For example, Canada has three markets—Vancouver, Toronto, and Montreal, which use the same currency, Canadian dollars.
Currency partition	Separates local currency members from the base currency defined in the application. This dimension type is used only in the main database and is only for currency conversion applications. The base currency for analysis may be U.S. dollars, and the local currency members may contain values that are based on the currency type of their region.

Member Storage Properties

You can specify data storage properties for members; data storage properties define where and when consolidations are stored. For example, by default, members are tagged as store data. Essbase sums the values of store data members and stores the result at the parent level.

You can change the default logic for each member by changing the data storage property tag for the member. For example, you can change a store data member to a label only member. Members with the label only tag, for example, do not have data associated with them.

Table 6 describes the effect that Essbase data storage properties have on members.

Table 6 Essbase Data Storage Properties

Data Storage Properties	Effects on Members
Store data	The member stores data. Store data is the default storage property.
Dynamic Calc	The data associated with the member is not calculated until requested by a user. The calculated data is not stored; it is discarded after the request is completed.
Dynamic Calc and Store	The data associated with the member is not calculated until it is requested by a user. The calculated data is then stored.
Shared member	The data associated with the member comes from another member with the same name.
Never share	The data associated with the member is duplicated with the parent and its child if an implied shared relationship exists.
Label only	Although a label only member has no data associated with it, it can display a value. The label only tag groups members and eases navigation and reporting. Typically, label only members are not calculated. For example, in the Measures dimension, the member Ratios has three children, Margin %, Profit%, and Profit per Ounce. The member Ratios defines a category of members. When consolidated, Margin%, Profit%, and Profit per Ounce do not roll up to a meaningful figure for Ratios. Hence, Ratios is tagged as label only.

Checklist for Dimension and Member Properties

- Can you identify a time dimension?
- Can you identify an accounts dimension?
- Does the data include foreign currencies? If so, did you identify a currency partition dimension?
- Can you identify qualities or characteristics of dimensions that should be defined as separate attribute dimensions?
- Which members require special data storage properties?

Designing an Outline to Optimize Performance

Position attribute dimensions at the end of the outline. Position dense dimensions before sparse dimensions.

The position of dimensions in an outline and the storage properties of dimensions can affect two areas of performance—how quickly calculations are run and how long it takes users to retrieve information.

Use the following topics to understand performance optimization basics.

Optimizing Query Performance

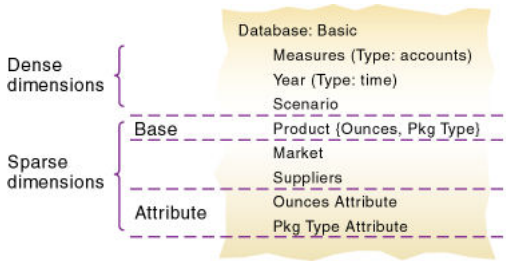
To optimize query performance, use the following guidelines when you design an outline:

- If the outline contains attribute dimensions, ensure that the attribute dimensions are the only sparse Dynamic Calc dimensions in the outline.
- In the outline, place the more-queried sparse dimensions before the less-queried sparse dimensions.

The outline in Figure 31 is designed for optimum query performance:

- Because the outline contains attribute dimensions, the storage property for standard dimensions and all standard dimensions members is set as store data.
- As the most-queried sparse dimension, the Product dimension is the first of the sparse dimensions. Base dimensions are typically queried more than other dimensions.

Figure 31 Designing an Outline for Optimized Query Times



Optimizing Calculation Performance

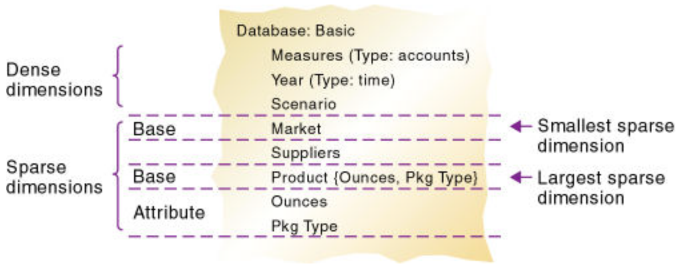
To optimize calculation performance, order the sparse dimensions in the outline by their number of members, starting with the dimension that contains the fewest.

See "Designing for Calculation Performance" on page 859.

The outline in Figure 32 is designed for optimum calculation performance:

- The smallest standard dimension that is sparse, Market, is the first of the sparse dimensions in the outline.
- The largest standard dimension that is sparse, Product, is immediately above the first attribute dimension. If the outline did not contain attribute dimensions, the Product dimension would be at the end of the outline.

Figure 32 Designing an Outline for Optimized Calculation Times



Meeting the Needs of Both Calculation and Retrieval

Although they contain the same dimensions, the example outlines in Figure 31 on page 95 and Figure 32 on page 96 are different. To determine the best outline sequence for a situation, prioritize the data retrieval requirements of the users against the time needed to run calculations on the database. How often do you expect to update and recalculate the database? What is the nature of user queries? What is the expected volume of user queries?

A possible workaround is initially to position the dimensions in the outline to optimize calculation. After you run the calculations, you can manually resequence the dimensions to optimize retrieval. When you save the outline after you reposition its dimensions, choose to restructure the database by index only. Before you run calculations again, resequence the dimensions in the outline to optimize calculation.

Checking System Requirements

Now you are ready to determine the system requirements for the database.

- Ensure that you have enough disk space.

  See "Determining Disk Space Requirements" on page 1020.
- Ensure that you have enough memory.

  See "Estimating Memory Requirements" on page 1033.

- Ensure that your caches are set correctly.

  See Chapter 52, “Optimizing Essbase Caches.”

Loading Test Data

Before you can test calculations, consolidations, and reports, you need data in the database. During the design process, loading mocked-up data or a subset of real data provides flexibility and shortens the time required to test and analyze results.

Detailed instructions for loading data are in the following chapters:

- Chapter 16, “Understanding Data Loading and Dimension Building”
- Chapter 17, “Working with Rules Files”
- Chapter 18, “Using a Rules File to Perform Operations on Records, Fields, and Data”
- Chapter 19, “Performing and Debugging Data Loads or Dimension Builds”
- Chapter 20, “Understanding Advanced Dimension Building Concepts”

If you are satisfied with your database design after the preliminary test, test load the complete set of real data with which you will populate the final database. Use the test rules files if possible. This final test may reveal source data problems that were not anticipated during earlier design process phases.

Defining Calculations

Calculations are essential to derive certain types of data. Data that is derived from a calculation is called calculated data; basic noncalculated data is called input data.

The following topics use the Product and Measures dimensions of the TBC application to illustrate several types of common calculations that are found in many Essbase databases.

For information on block storage calculations, see the following chapters:

- Chapter 21, “Calculating Essbase Databases”
- Chapter 31, “Developing Custom-Defined Calculation Functions”
- Chapter 25, “Understanding Intelligent Calculation”

Consolidation of Dimensions and Members

When you define members of standard dimensions, Essbase automatically tags the members with the + (plus sign representing addition) consolidator, meaning that during consolidation members are added. As appropriate, you can change a member consolidation property to one of the following operators: - (dash representing subtraction), * (asterisk representing multiplication), / (forward slash representing division), % (percent sign representing percentage), and ~ (tilda representing no consolidation).

Consolidation is the most frequently used calculation in Essbase. This topic uses the Product dimension to illustrate consolidations.

The TBC application has several consolidation paths:

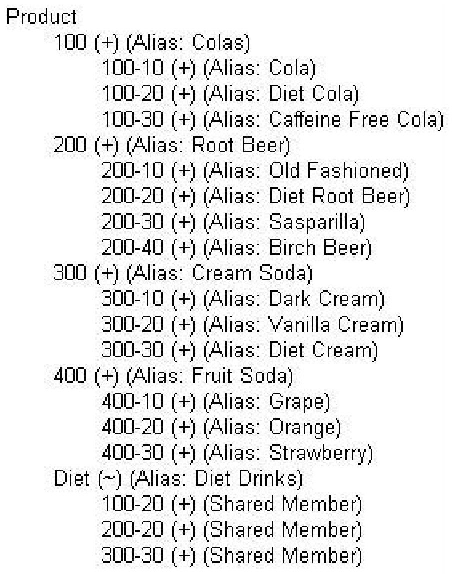
- Individual products roll up to product families, and product families consolidate into Product. The TBC outline also requires multiple consolidation paths; some products must consolidate in multiple categories.
- States roll up to regions, and regions consolidate into Market.
- Months roll up into quarters, and quarters consolidate into Year.

The following topics discuss consolidation in greater detail:

- "Effect of Position and Operator on Consolidation" on page 99
- "Consolidation of Shared Members" on page 99
- "Checklist for Consolidation" on page 99

Consolidation operators define how Essbase rolls up data for each member in a branch to the parent. For example, using the default addition (+) operator, Essbase adds 100-10, 100-20, and 100-30 and stores the result in their parent, 100, as shown in Figure 33.

Figure 33 TBC Product Dimension



```
Product
    100 (+) (Alias: Colas)
        100-10 (+) (Alias: Cola)
        100-20 (+) (Alias: Diet Cola)
        100-30 (+) (Alias: Caffeine Free Cola)
    200 (+) (Alias: Root Beer)
        200-10 (+) (Alias: Old Fashioned)
        200-20 (+) (Alias: Diet Root Beer)
        200-30 (+) (Alias: Sasparilla)
        200-40 (+) (Alias: Birch Beer)
    300 (+) (Alias: Cream Soda)
        300-10 (+) (Alias: Dark Cream)
        300-20 (+) (Alias: Vanilla Cream)
        300-30 (+) (Alias: Diet Cream)
    400 (+) (Alias: Fruit Soda)
        400-10 (+) (Alias: Grape)
        400-20 (+) (Alias: Orange)
        400-30 (+) (Alias: Strawberry)
    Diet (~) (Alias: Diet Drinks)
        100-20 (+) (Shared Member)
        200-20 (+) (Shared Member)
        300-30 (+) (Shared Member)
```

The Product dimension contains mostly addition (+) operators, which indicate that each group of members is added and rolled up to the parent. Diet has a tilde (~) operator, which indicates that Essbase does not include the Diet member in the consolidation to the parent, Product. The Diet member consists entirely of members that are shared. The TBC product management group wants to be able to isolate Diet drinks in reports, so TBC created a separate Diet member that does not impact overall consolidation.

Effect of Position and Operator on Consolidation

Essbase calculates the data of a branch in top-down order. For example, if you have, in order, two members tagged with an addition (+) operator and a third member tagged with a multiplication (*) operator, Essbase adds the first two and multiplies that sum by the third.

Because Essbase always begins with the top member when it consolidates, the order and the labels of the members is important. See "Calculating Members with Different Operators" on page 146.

Table 7 defines Essbase consolidation operators.

Table 7 Consolidation Operations

Consolidation Operator	Description
+	The default operator. Essbase adds the member to the result of previous calculations performed on members of the branch.
–	Essbase multiplies the member by -1 and then adds the product to the result of previous calculations performed on members of the branch.
*	Essbase multiplies the member by the result of previous calculations performed on members of the branch.
/	Essbase divides the member into the result of previous calculations performed on members of the branch.
%	Essbase divides the member into the sum of previous calculations performed on members of the branch. The result is multiplied by 100.
~	Essbase does not use the value of the member in the consolidation to its parent.

Consolidation of Shared Members

Shared members also affect consolidation paths. The shared member concept enables two members with the same name to share the same data. The shared member stores a pointer to data contained in the other member, so Essbase stores the data only once. Shared members must be in the same dimension. Data can be shared by multiple members.

Checklist for Consolidation

Use the following checklist to help define consolidation:

- Did you identify the consolidations in the outline?
- Did you tag each member with the proper consolidation operator?
- Did you specify a shared member tag for designated members?
- Would shared members be more efficient if designed within an attribute dimension (other than shared)?

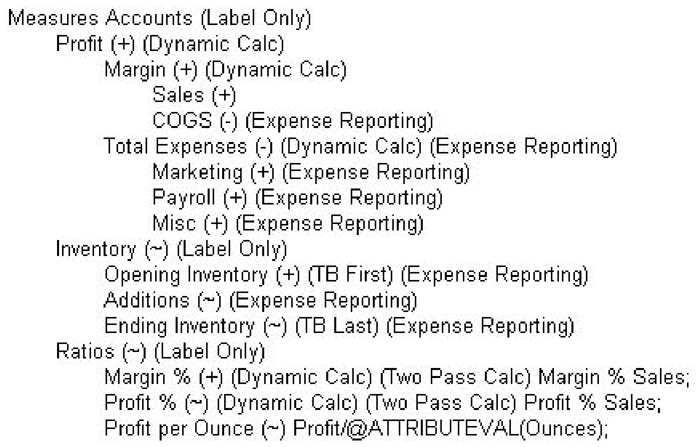
Tags and Operators on Example Measures Dimension

The Measures dimension is the most complex dimension in the TBC outline because it uses both time and accounts data. It also contains formulas and special tags to help Essbase calculate the outline. This topic discusses the formulas and tags that TBC included in the Measures dimension (the dimension tagged as accounts).

Look closely at the Measures dimension tags defined by TBC (in Figure 34). Many of the properties of the Measures dimension are discussed in previous topics of this chapter: addition (+), subtraction (–), and no consolidation (~) operators, and accounts and label only tags:

- The Inventory and Ratios member names assist the user in data navigation. They do not contain data and therefore receive a label only tag.
- The Measures dimension itself has a label only tag. Some members of Measures have a Dynamic Calc tag. Dynamic calculations are discussed in "Dynamic Calculations" on page 103.
- Some members of Measures have a time balance tag (TB First or TB Last). Time balance tags are discussed in "Setting Time Balance Properties" on page 142.

Figure 34 TBC Measures Dimension



```
Measures Accounts (Label Only)
    Profit (+) (Dynamic Calc)
        Margin (+) (Dynamic Calc)
            Sales (+)
            COGS (-) (Expense Reporting)
        Total Expenses (-) (Dynamic Calc) (Expense Reporting)
            Marketing (+) (Expense Reporting)
            Payroll (+) (Expense Reporting)
            Misc (+) (Expense Reporting)
    Inventory (~) (Label Only)
        Opening Inventory (+) (TB First) (Expense Reporting)
        Additions (~) (Expense Reporting)
        Ending Inventory (~) (TB Last) (Expense Reporting)
    Ratios (~) (Label Only)
        Margin % (+) (Dynamic Calc) (Two Pass Calc) Margin % Sales;
        Profit % (~) (Dynamic Calc) (Two Pass Calc) Profit % Sales;
        Profit per Ounce (~) Profit/@ATTRIBUTEVAL(Ounces);
```

Accounts Dimension Calculations

This topic discusses two forms of calculations for a dimension tagged as accounts, time balance properties and variance reporting.

Time Balance Properties

Note the two tags in the Measures dimension of Table 9—TB first and TB last. These tags, called time balance tags or properties, provide instructions to Essbase about how to calculate the data in a dimension tagged as accounts. Using the tags requires a dimension tagged as accounts and a dimension tagged as time. The first, last, average, and expense tags are available exclusively for use with accounts dimension members.

In the TBC Measures dimension, Opening Inventory data represents the inventory that TBC carries at the beginning of each month. The quarterly value for Opening Inventory equals the Opening value for the quarter. Opening Inventory requires the time balance tag, TB first.

Ending Inventory data represents the inventory that TBC carries at the end of each month. The quarterly value for Ending Inventory equals the ending value for the quarter. Ending Inventory requires the time balance tag, TB last. Table 8 defines the time balance tags for the accounts dimension.

Table 8 Accounts Member Tags

Tags	Description
Time Balance Last	The value for the last child member is carried to the parent. For example, March is carried to Qtr1.
Time Balance First	The value for the first child is carried to the parent. For example, Jan is carried to Qtr1.

In Table 9, Qtr1 (second column from the right) and Year (right-most column) show how consolidation in the time dimension is affected by time balance properties in the accounts dimension. Data is shown for the first quarter only.

Table 9 TBC Consolidations Affected by Time Balance Properties

Dimensions	Jan	Feb	Mar	Qtr1	Year
Accounts Member1	11	12	13	*36*	*Qtr1 + Qtr2 + Qtr3 + Qtr4*
Accounts Member2 (TB First)	20	25	21	*20*	*20*
Accounts Member3 (TB Last)	25	21	30	*30*	*Value of Qtr4*

Normally, the calculation of a parent in the time dimension is based on the consolidation and formulas of children of the parent. However, if a member in an accounts branch is marked as TB First, any parent in the time dimension matches the member marked as TB First.

For examples, see "Setting Time Balance Properties" on page 142.

Variance Reporting

One TBC Essbase requirement is the ability to perform variance reporting on actual versus budget data. The variance reporting calculation requires that any item that represents an expense to the company must have an expense reporting tag. Inventory members, Total Expense members, and the COGS member each receive an expense reporting tag for variance reporting.

Essbase provides two variance reporting properties—expense and nonexpense. The default is nonexpense. Variance reporting properties define how Essbase calculates the difference between actual and budget data in members with the @VAR or @VARPER function in their member formulas.

When you tag a member as expense, the @VAR function calculates Budget – Actual. For example, if the budgeted amount is $100 and the actual amount is $110, the variance is –10.

Without the expense reporting tag, the @VAR function calculates Actual – Budget. For example, if the budgeted amount is $100 and the actual amount is $110, the variance is 10.

Formulas and Functions

Formulas calculate relationships between members in the database outline. You can apply formulas to members in the outline, or you can place formulas in a calculation script. This topic explains how TBC optimized the performance of its database by using formulas.

Functions are predefined routines that perform specialized calculations and return sets of members or sets of data values. Formulas comprise operators and functions, as well as dimension names, member names, and numeric constants.

Essbase supports the following operators:

- Mathematical operators that perform arithmetic operations
- Conditional operators that build logical conditions into calculations
- Cross-dimensional operators that point to data values of specific database member combinations

The Essbase functions include more than 100 predefined routines to extend the calculation capabilities of Essbase. Essbase supports the following functions:

- Boolean functions, which provide a conditional test by returning a TRUE or FALSE value
- Mathematical functions, which perform specialized mathematical calculations
- Relationship functions, which look up data values within a database during a calculation based on the position of the current member
- Range functions, which declare a range of members as an argument to another function or to a command
- Financial functions, which perform specialized financial calculations
- Member set functions, which are based on a specified member and which generate lists of members
- Allocation functions, which allocate values that are input at a parent level across child members
- Forecasting functions, which manipulate data for the purpose of smoothing data, interpolating data, or calculating future values
- Statistical functions, which calculate advanced statistics
- Date and time functions, which use date and time characteristics in calculation formulas
- Calculation mode functions, which specify the calculation mode that Essbase uses to calculate a formula

The Measures dimension uses the following formulas:

- Margin = Sales – COGS
- Total Expenses = Marketing + Payroll + Miscellaneous

- Profit = Margin – Total Expenses
- Profit % = Profit % Sales
- Margin % = Margin % Sales
- Profit per Ounce = Profit / @ATTRIBUTEVAL(@NAME(Ounces))

Essbase uses consolidation operators to calculate the Margin, Total Expenses, and Profit members. The Margin% formula uses a % operator, which means "express Margin as a percentage of Sales." The Profit% formula uses the same % operator. The Profit per Ounce formula uses a division operator (/) and a function (@ATTRIBUTEVAL) to calculate profitability by ounce for products sized in ounces.

Note:

In the Profit per Ounce formula, the @NAME function is also used to process the string "Ounces" for the @ATTRIBUTEVAL function.

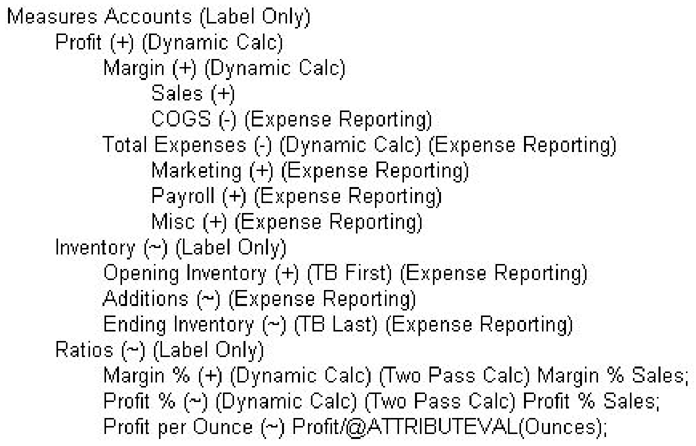
For a complete list of operators, functions, and syntax, see the *Oracle Essbase Technical Reference*. Also see Chapter 22, "Developing Formulas".

Dynamic Calculations

When you design the overall database calculation, you may want to define a member as a Dynamic Calc member. When you tag a member as Dynamic Calc, Essbase calculates the combinations of that member when you retrieve the data, instead of precalculating the member combinations during the regular database calculation. Dynamic calculations shorten regular database calculation time but may increase retrieval time for dynamically calculated data values.

In Figure 35, the TBC Measures dimension contains several members tagged as Dynamic Calc —Profit, Margin, Total Expenses, Margin %, and Profit %.

Figure 35 TBC Measures Dimension, Dynamic Calc Tags



```
Measures Accounts (Label Only)
    Profit (+) (Dynamic Calc)
        Margin (+) (Dynamic Calc)
            Sales (+)
            COGS (-) (Expense Reporting)
        Total Expenses (-) (Dynamic Calc) (Expense Reporting)
            Marketing (+) (Expense Reporting)
            Payroll (+) (Expense Reporting)
            Misc (+) (Expense Reporting)
    Inventory (~) (Label Only)
        Opening Inventory (+) (TB First) (Expense Reporting)
        Additions (~) (Expense Reporting)
        Ending Inventory (~) (TB Last) (Expense Reporting)
    Ratios (~) (Label Only)
        Margin % (+) (Dynamic Calc) (Two Pass Calc) Margin % Sales;
        Profit % (~) (Dynamic Calc) (Two Pass Calc) Profit % Sales;
        Profit per Ounce (~) Profit/@ATTRIBUTEVAL(Ounces);
```

When an overall database calculation is performed, the Dynamic Calc members and their corresponding formulas are not calculated. These members are calculated when a user requests them, for example, from Spreadsheet Add-in. Essbase does not store the calculated values; it recalculates the values for any subsequent retrieval. However, you can choose to store dynamically calculated values after the first retrieval.

To decide when to calculate data values dynamically, consider your priorities in the following areas:

- Optimum regular calculation time (batch calculation)
- Low disk space usage
- Reduced database restructure time
- Speedy data retrieval for users
- Reduced backup time

See Chapter 26, "Dynamically Calculating Data Values".

Two-Pass Calculations

In the TBC database, Margin % and Profit % contain the label two-pass. This default label indicates that some member formulas must be calculated twice to produce the desired value. The two-pass property works only on members of the dimension tagged as accounts and on members tagged as Dynamic Calc and Dynamic Calc and Store.

The following example illustrates why Profit % (based on the formula Profit % Sales) has a two-pass tag. The tables have five columns (column headers are labeled left to right as Dimension, Jan, Feb, Mar, and Qtr1) and three rows (labeled as Profit, Sales, and Profit %). Jan, Feb, Mar, and Qtr1 are members of the Year dimension. Profit, Sales, and Profit % are members of the Measures (accounts) dimension.

Table 10 on page 104 defines the initial data to load into Essbase. The data values for Profit -> Jan, Profit -> Feb, and Profit -> Mar are 100. The data value for Sales -> Jan, Sales -> Feb, and Sales -> Mar are 1000.

Table 10 Data Loaded into Essbase

Dimension	Jan	Feb	Mar	Qtr1
Profit	100	100	100	
Sales	1000	1000	1000	
Profit %				

First, Essbase calculates the Measures dimension. In Table 11 on page 104, the data values for Profit % -> Jan, Profit % -> Feb, and Profit % -> Mar are 10%.

Table 11 Data After Essbase Calculates the Measures Dimension

Dimension	Jan	Feb	Mar	Qtr1
Profit	100	100	100	

Dimension	Jan	Feb	Mar	Qtr1
Sales	1000	1000	1000	
Profit %	10%	10%	10%	

Next, Essbase calculates the Year dimension. The data rolls up across the dimension. In Table 12 on page 105, the data values for Profit -> Qtr1 (300) and Sales -> Qtr1 (3000) are correct. The data value for Profit % -> Qtr1 (30%) is incorrect because Profit % is tagged as a two-pass calculation.

Table 12 Data After Essbase Calculates the Year Dimension

Dimension	Jan	Feb	Mar	Qtr1
Profit	100	100	100	300
Sales	1000	1000	1000	3000
Profit %	10%	10%	10%	30%

Essbase then recalculates profit percentage at each occurrence of the member Profit %. In Table 13 on page 105, the data value for Profit % -> Qtr1 (10%) is correct after the second pass.

Table 13 Data After Essbase Recalculates Profit Percentage

Dimension	Jan	Feb	Mar	Qtr1
Profit	100	100	100	300
Sales	1000	1000	1000	3000
Profit %	10%	10%	10%	10%

Checklist for Calculations

Use the following checklist when you define a calculation:

- Does the default calculation logic achieve accurate results?
- Which members require formulas?
- Which members require time balance tags?
- Which members require variance reporting?
- Which members require two-pass calculation?
- Which members can be tagged as Dynamic Calc?

Note:

The Essbase triggers feature enables efficient monitoring of data changes in a database. See "Understanding Triggers Definitions" on page 119.

Defining Reports

To ensure that the design meets user information requirements, you must view data as users view it. Users typically view data through spreadsheets, printed reports, or reports published on the Web. Oracle and its partners offer many tools for producing the reporting systems that users use.

Several tools can help you display and format data quickly, and test whether the database design meets user needs. You can use the Report Script Editor in Administration Services Console to write report scripts quickly. Those familiar with spreadsheets can use the Spreadsheet Add-in or Smart View (Smart View requires Provider Services).

During the design phase, check for the following things:

- Grouping and sequencing of data. Do the intersections enabled by the design provide the data that users need?
- Levels of totals. What consolidation levels are required by, for example, a Spreadsheet Add-in user who drills down and up through the hierarchy of the outline design?
- Attribute reporting. Does the database design facilitate an analysis that is based on the characteristics or attributes of specific dimensions or members? For example, do you need to compare sales by specific combinations of size and packaging, such as comparing the sales of 16-ounce bottled colas with the sales of 32-ounce bottled colas?

Be sure to use the appropriate tool to create and test predesigned use reports against the test data. The reports that you design should provide information that meets your original objectives. The reports should be easy to use, providing the right combinations of data and the right amount of data. Because reports with too many columns and rows are difficult to use, you may need to create several reports instead of one all-inclusive report.

Verifying the Design

After you analyze the data and create a preliminary design, check all aspects of the design with users. You should already have verified that the database satisfies the users' analysis and reporting needs. Ensure that the database satisfies all of their goals.

Do the calculations provide the information they need? Are they able to generate reports quickly? Are they satisfied with consolidation times? In short, ask users if the database works for them.

Near the end of the design cycle, test with real data. Does the outline build correctly? Does all data load? If the database fails in any area, repeat the steps of the design cycle to identify the cause of the problem.

Essbase provides several sources of information to help you isolate problems. Sources include application and Essbase Server logs, exception logs, and database information accessible from Administration Services. Look at documentation topics relevant to your problem; for example, topics about security, calculations, reports, or general error messages. Use the index of this guide for help in solving problems. Look up such terms as troubleshooting, logs, optimizing, performance, recovery, resources, errors, and warnings.

Most likely, you will need to repeat one or more steps of the design process to arrive at the ideal database solution.

5 About Administration Services

In This Chapter

Introduction

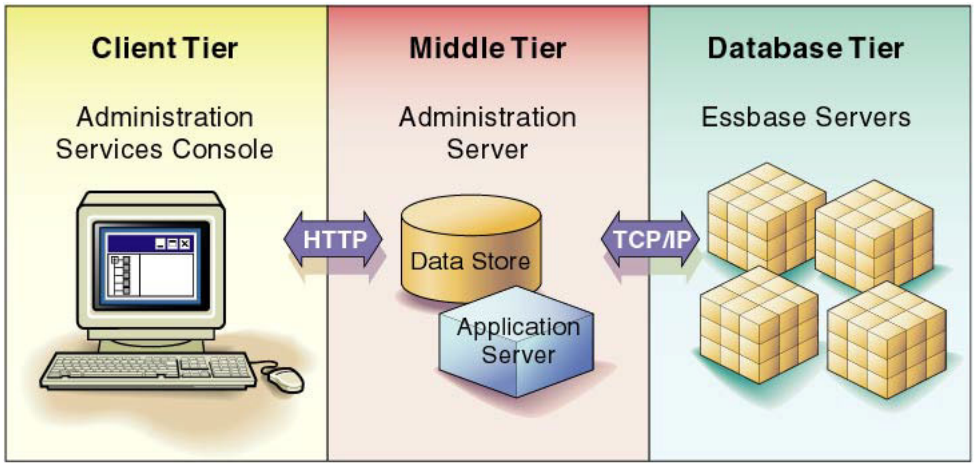
Administration Services is the cross-platform administration tool for Essbase. Administration Services consists of a Java middle-tier server, called Essbase Administration Server, and a Java client console, called Administration Services Console.

Administration Services Console is the graphical user interface (GUI) that enables administrators to manage the Essbase environment from one navigation tree, called Enterprise View. The console provides wizards, editors, and other tools to help administrators view, manage, and maintain a unique set of Essbase Servers. The console includes a data preview grid that enables you to preview data without having to switch from the console to another program.

Administration Services Architecture

Administration Services works with Essbase Servers in a three-tiered system consisting of a client user interface (UI), a middle-tier server, and one or more Essbase Servers. The middle tier coordinates interactions and resources between the UI and Essbase Servers. The tiers, which may or may not be on the same computer or platform, include the following components, as illustrated below:

Figure 36   Administration Services Architecture



- Client tier (Administration Services Console): A Java-based client console provides a UI to manage the Essbase environment.
- Middle tier (Essbase Administration Server): A Java-based server maintains communication, session, and security information for connections to Essbase Servers.
- Database tier (Essbase Server): One or more Essbase Servers store and process multidimensional database information. Essbase Servers are installed separately from Administration Services.

Deploying Administration Services

Administration Services can be deployed in a variety of scenarios. For example, you can install Essbase Server on a computer running UNIX and install Essbase Administration Server and Administration Services Console on a computer running Windows, or you can install Essbase Administration Server and Administration Services Console on separate computers and platforms. The middle tier Essbase Administration Server also supports the substitution of certain third-party products within the existing framework (for example, application servers and relational databases).

See the *Oracle Hyperion Enterprise Performance Management System Installation and Configuration Guide*.

Starting Administration Services

To start Administration Services, start Essbase Administration Server, and then start Administration Services Console. See “Starting Administration Services” in the *Oracle Essbase Administration Services Online Help*.

If you have enabled an Essbase Server for remote start, you can start that Essbase Server remotely from Enterprise View in Administration Services Console. To enable this functionality, configure and start the Remote Start Service on the Essbase Server computer. See “Starting Essbase Server Remotely from Administration Services Console” on page 694.

About Administration Services Users

Existing Essbase users cannot use Administration Services until they have also been created as users on Essbase Administration Server. You can use the User Setup Wizard to step you through the process of creating Essbase Administration Server users and associating them with the appropriate Essbase Servers. You do not need to create Excel spreadsheet users on Essbase Administration Server.

➤ To create Administration Services users, see "User Setup Wizard" in the *Oracle Essbase Administration Services Online Help*.

Connecting to Administration Services

In Administration Services, connections to individual Essbase Servers are handled by the middle tier Essbase Administration Server. When you start Administration Services, if Essbase Server is started, you are automatically connected to each Essbase Server you added to Enterprise View. See "About Essbase Connections and Ports" in the *Oracle Essbase Administration Services Online Help*.

You can connect to different instances of Essbase Server simultaneously from Administration Services Console.

Your Administration Services user name and password may be different than your Essbase Server user name and password. If you do not know your Administration Services user name and password, see your Administration Services administrator.

After your initial connection to Administration Services, you can use the User Setup Wizard to create Administration Services users and add Essbase Servers to each user's Enterprise View. See "Connecting to Administration Services" in the *Oracle Essbase Administration Services Online Help*.

Adding Essbase Administration Servers to Enterprise View

Each time you connect to Administration Services, the Essbase Administration Servers you have chosen are displayed in Enterprise View, which is the navigation tree in the left navigation pane. Each user can populate Enterprise View with a unique set of Essbase Administration Servers.

➤ To add Essbase Administration Servers to Enterprise View, see "Adding Essbase Administration Servers to Enterprise View" in the *Oracle Essbase Administration Services Online Help*.

Adding Essbase Servers to Enterprise View

Each time you connect to Administration Services, the Essbase Servers you have chosen are displayed in Enterprise View, which is the navigation tree in the left navigation pane. Each user can populate Enterprise View with a unique set of Essbase Servers. You can use the following methods to add Essbase Servers to Enterprise View:

- In Enterprise View, right-click an Essbase Server node, and select Add Essbase Server.
- Select File, then Wizards, then User Setup. Follow the wizard instructions.
- For an existing server, edit the server's properties.

You can also create custom views of the Enterprise View tree in separate tabs in the navigation pane. See "About Custom Views" in the *Oracle Essbase Administration Services Online Help*.

➤ To add Essbase Servers to Enterprise View, see "Adding Essbase Servers to Enterprise View" in the *Oracle Essbase Administration Services Online Help*.

About Essbase Server Connections and Ports

The number of ports available for an Essbase Server represents the number of licensed concurrent connections. Essbase provides one reserve port for the system administrator. A system administrator uses the reserve port to log out one or more users when all other ports are in use. See Chapter 43, "Running Essbase Servers, Applications, and Databases."

In Administration Services, a port is in use only when an Essbase Server connection is established. See "About Essbase Connections and Ports" in the *Oracle Essbase Administration Services Online Help*.

About Essbase Administration Server

The middle tier Essbase Administration Server provides business logic to support cross-server operations, persistence of user preferences, and access to Essbase Servers. A system administrator creates users on Essbase Administration Server, and then Essbase Administration Server manages their connections to Essbase.

In Enterprise View, the node name for Essbase Administration Server is the same as the server computer name.

Essbase Administration Server has several configurable communication ports, which are different from Essbase Server ports. If one of the default communication ports is in use by another application, you must specify another port value in order to run Essbase Administration Server.

Note:

If you change the value for the Essbase Administration Server port, you must specify the new port value when you log in to the Administration Services Console.

➤ To change a default port value, see "Specifying Communication Ports for Administration Server" in the *Oracle Essbase Administration Services Online Help*.

Part II

Designing and Creating Applications and Databases

In Designing and Creating Applications and Databases:

6

Creating Applications and Databases

In This Chapter

Some information in this chapter applies only to block storage databases and is not relevant to aggregate storage databases.

Also see:

- Chapter 57, “Comparison of Aggregate and Block Storage”
- Chapter 58, “Aggregate Storage Applications, Databases, and Outlines”

Process for Creating Applications and Databases

➤ To create an application and database:

1 **Design the application.**

See “Quick Start for Implementing Essbase” on page 57.

2 **Create an application.**

See “Creating an Application” on page 119.

3 **Create a database.**

See “Creating a Database” on page 119.

4 **If necessary, set substitution variables at the Essbase Server, application, or database level.**

See “Using Substitution Variables” on page 120.

5 **If necessary, set a location alias for the database.**

See “Using Location Aliases” on page 124.

6 **Create the outline.**

See Chapter 7, “Creating and Changing Database Outlines.”

Understanding Applications and Databases

An Essbase application is a management structure that contains one or more Essbase databases and related files. Essbase applications and databases reside on an Essbase Server. The server computer can store multiple applications.

An Essbase database is a data repository that contains a multidimensional data storage array. A multidimensional database supports multiple views of data so that users can analyze the data and make meaningful business decisions. See “Understanding Multidimensional Databases” on page 61 and “Storage Allocation” on page 767.

Understanding Database Artifacts

Files that are related to databases are called artifacts (or objects). Database artifacts perform actions against one or more Essbase databases, such as defining calculations or reporting against data. By default, artifacts are stored in their associated database folder on the Essbase Server, and can also be saved to a client computer or to other available network directories. You cannot, however, load data or calculate data on a client computer.

Essbase provides the following common types:

- A database outline (a storage structure definition)
- Data sources
- Rules for loading data and building dimensions dynamically (rules files)
- Scripts that define how to calculate data (calculation scripts)
- Scripts that generate reports on data (report scripts)
- Security definitions
- Linked reporting objects (LROs)
- Partition definitions

Some of these artifacts are optional, such as calculation scripts and LROs. See “Application and Database File Types” on page 711.

In Administration Services Console, database artifacts are displayed under their associated applications or database in the Enterprise View tree.

Understanding Database Outlines

Database outlines define the structure of a multidimensional database, including all the dimensions, members, aliases, properties, types, consolidations, and mathematical relationships. The structure defined in the outline determines how data is stored in the database.

When a database is created, Essbase creates an outline for that database automatically. The outline has the same name as the database (`dbname.otl`). For example, when the Basic database is created within the Sample application, an outline is created in the following directory:

```
ARBORPATH/app/sample/basic/basic.otl
```

See "Creating a Database" on page 119 and Chapter 7, "Creating and Changing Database Outlines."

Understanding Data Sources

A data source is external data that is loaded into an Essbase database. The common types of data sources include the following:

- Text files
- Spreadsheet files
- Spreadsheet audit log files
- External databases, such as an SQL database

See "Supported Data Sources" on page 258.

Understanding Rules Files for Data Load and Dimension Build

An Essbase database contains no data when it is created. Data load rules files are sets of operations that Essbase performs on data from an external data source file as the data is loaded, or copied, into the Essbase database. Dimension build rules files create or modify the dimensions and members in an outline dynamically based on data in an external data source. Rules files are typically associated with a particular database, but you can define rules for use with multiple databases. One rules file can be used for both data loads and dimension builds. Rules files have a `.rul` extension.

See "Rules Files" on page 263 and Chapter 17, "Working with Rules Files."

Understanding Calculation Scripts

Calculation scripts are text files that contain sets of instructions telling Essbase how to calculate data in the database. Calculation scripts perform calculations different from the consolidations and mathematical operations that are defined in the database outline. Because calculation scripts perform specific mathematical operations on members, they are typically associated with a particular database. You can, however, define a calculation script for use with multiple databases. Calculation scripts files have a `.csc` extension.

See Chapter 28, "Developing Calculation Scripts."

Understanding Report Scripts

Report scripts are text files that contain data retrieval, formatting, and output instructions to create a report from the database. Report scripts are typically associated with a particular database, but you can define a report script for use with multiple databases. Report scripts have a `.rep` extension.

See Chapter 33, "Developing Report Scripts."

Understanding Security Definitions

Essbase provides a comprehensive system for managing access to applications, databases, and other artifacts. Each application and database contains its own security definitions that restrict user access.

See Chapter 38, "User Management and Security."

Understanding LROs

An LRO is an artifact associated with a specific data cell in an Essbase database. LROs can enhance data analysis capabilities by providing additional information on a cell.

An LRO can be any of the following:

- A paragraph of descriptive text (a "cell note")
- A separate file that contains text, audio, video, or graphics
- A URL for a Web site
- A link to data in another Essbase database

See Chapter 11, "Linking Objects to Essbase Data."

Understanding Spreadsheet Queries

Within Spreadsheet Add-in, users can create and save queries using Query Designer (EQD). Users with access to the query can access it later. Query files created using Query Designer have a `.eqd` extension.

See the *Oracle Essbase Spreadsheet Add-in User's Guide*.

Understanding Member Select Definitions

Within Spreadsheet Add-in, users can define and save member retrievals with the member select feature. Member specification files have a `.sel` extension.

See the *Oracle Essbase Spreadsheet Add-in User's Guide*.

Understanding Triggers Definitions

The triggers feature enables efficient monitoring of data changes in a database. If data breaks rules that you specify in a trigger, Essbase logs relevant information in a file or, for some triggers, sends an e-mail alert; for example, to notify the sales manager if, in the Western region, sales for a month fall below sales for the equivalent month in the previous year.

See "Monitoring Data Changes Using Triggers" on page 725.

Creating Applications and Databases

Create an application and then create its databases. You can annotate the databases.

Creating an Application

When you create an application on the Essbase Server, Essbase creates a subdirectory for the application on the Essbase Server in the *ARBORPATH*`/app` directory. The new subdirectory has the same name as the application; for example, *ARBORPATH*`/app/app1`. In Administration Services Console, applications and databases are displayed in a tree structure in Enterprise View.

Before naming the application, see "Naming Restrictions for Applications and Databases" on page 1059.

You can also create an application that is a copy of an existing application. See "Copying or Migrating Applications" on page 716.

➤ To create an application, use a tool:

Tool	Topic	Location
Administration Services	Creating Applications	*Oracle Essbase Administration Services Online Help*
MaxL	**create application**	*Oracle Essbase Technical Reference*
ESSCMD	CREATEAPP	*Oracle Essbase Technical Reference*

Creating a Database

When you create a database, Essbase creates a subdirectory for the database within the application directory. The new subdirectory has the same name as the database; for example, *ARBORPATH*`/app/app1/dbname1`. In Administration Services Console, applications and databases are displayed in a tree structure in Enterprise View.

You can create normal databases or currency databases. See Chapter 13, "Designing and Building Currency Conversion Applications."

Before naming the database, see "Naming Restrictions for Applications and Databases" on page 1059.

➤ To create a database, use a tool:

Tool	Topic	Location
Administration Services	Creating Databases	*Oracle Essbase Administration Services Online Help*
MaxL	**create database**	*Oracle Essbase Technical Reference*
ESSCMD	CREATEDB	*Oracle Essbase Technical Reference*

Except for databases requiring use of the Currency Conversion option, creating one database per application is recommended.

Annotating a Database

A database note can provide useful information when you need to broadcast messages to users about the status of a database, deadlines for updates, and so on. Users can view database notes in Spreadsheet Add-in.

➤ To annotate a database, see "Annotating Databases" in the *Oracle Essbase Administration Services Online Help*.

Using Substitution Variables

Substitution variables are global placeholders for regularly changing information. Because changes to a variable value are reflected everywhere the variable is used, manual changes are reduced.

For example, many reports depend on reporting periods; if you generate a report based on the current month, you must update the report script manually every month. With a substitution variable, such as `CurMnth`, set on the server, you can change the assigned value each month to the appropriate time period. When you use the variable name in a report script, the information is dynamically updated when you run the final report.

You can use substitution variables with both aggregate storage and block storage applications in the following areas:

- Aggregate storage outline formulas

  See "Using MDX Formulas" on page 931.

- Block storage outline formulas

  See "Using Substitution Variables in Formulas" on page 367.

- Calculation scripts (block storage databases only)

  See Chapter 28, "Developing Calculation Scripts."

- Report scripts

  See Chapter 33, "Developing Report Scripts."

- Data load rules file header definitions and field definitions. You can enter variable names for dimension and member names.

  See these *Oracle Essbase Administration Services Online Help* topics: “Setting Headers in the Rules File” and “Mapping Field Names.”
- Partition definitions

  See “Substitution Variables in Partition Definitions” on page 215.
- Data source name (DSN) specifications in rules files for SQL data sources

  See *Oracle Essbase SQL Interface Guide*.
- SELECT, FROM, or WHERE clauses in rules files for SQL data sources

  See *Oracle Essbase SQL Interface Guide*.
- Security filters

  See “Filtering Using Substitution Variables” on page 651.
- MDX statements

  See “Using Substitution Variables in MDX Queries” on page 569.
- Spreadsheet Add-in.

  See the *Oracle Essbase Spreadsheet Add-in User's Guide*.

You can set substitution variables on the Essbase Server using Administration Services, MaxL, or ESSCMD. Set the variable at any of the following levels:

- Essbase Server—providing access to the variable from all applications and databases on the Essbase Server
- Application—providing access to the variable from all databases within the application
- Database—providing access to the variable within the specified database

Rules for Setting Substitution Variable Names and Values

The following rules apply to substitution variable names and values:

- The substitution variable name must comprise alphanumeric characters or underscores (_) and cannot exceed the limit specified in Appendix A, “Limits.”
- The substitution variable name cannot include nonalphanumeric characters, such as hyphens (-), asterisks (*), and slashes (/). Do not use spaces, punctuation marks, or brackets ([ ]) in substitution variable names used in MDX.
- If substitution variables with the same name exist at server, application, and database levels, the order of precedence for the variables: a database-level substitution variable supersedes an application-level variable, which supersedes a server-level variable.
- The substitution variable value may contain any character except a leading ampersand (&). The substitution variable value cannot exceed the limit specified in Appendix A, “Limits.”

- To set a substitution variable value to a duplicate member name, use the qualified member name enclosed in double quotation marks; for example, a value for &Period could be "[2006].[Qtr1]".
- When specifying use of a substitution variable, do not insert a substitution variable as a part of a qualified name. For example, it is invalid to specify "[2004].[&CurrentQtr]".
- If a substitution variable value is a member name that begins with a numeral or contains spaces or any of the special characters listed in "Using Dimension and Member Names in Calculation Scripts, Report Scripts, Formulas, Filters, Substitution Variable Values and Environment Variable Values" on page 1063, different rules apply for how you enter the variable:
  - Enclose the member-name value in brackets ([ ]) if it is used in MDX statements.
  - Enclose the member-name value in quotation marks (" ") if it is not used in MDX statements.
- If a substitution variable value is numeric, different rules apply for how you enter the variable:
  - If it is not used in MDX statements, enclose a substitution variable value in quotation marks; for example, if the variable name is Month, and its corresponding value is 01 (corresponding to January), place quotation marks around 01 ("01"). Substitution variables usually are used with block storage databases; they are not used in MDX statements.
  - If it is used in MDX statements only, such as in formulas in aggregate storage outlines, and the value is numeric or a member name, do not enclose the value in quotation marks.

Note:

If a substitution variable value is numeric or a member name starting with a numeral or containing the special characters referred to above is to be used both in MDX and non-MDX situations, create two substitution variables, one without the value enclosed in quotation marks and one with the value in quotation marks.

Setting Substitution Variables

You can set substitution variables on the Essbase Server at the server, application, or database level. Before setting a substitution variable, see "Rules for Setting Substitution Variable Names and Values" on page 121.

➤ To set a substitution variable, use a tool:

Tool	Topic	Location
Administration Services	Managing Substitution Variables	*Oracle Essbase Administration Services Online Help*
MaxL	**alter system**	*Oracle Essbase Technical Reference*

Tool	Topic	Location
	alter application **alter database**	
ESSCMD	CREATEVARIABLE	*Oracle Essbase Technical Reference*

To ensure that a new substitution variable value is available in formulas, partition definitions, and security filters, stop and restart the application. All other uses of substitution variables are dynamically resolved when used.

Deleting Substitution Variables

You may need to delete a substitution variable that is no longer used.

➤ To delete a substitution variable, use a tool:

Tool	Topic	See
Administration Services	Managing Substitution Variables	*Oracle Essbase Administration Services Online Help*
MaxL	**alter system** **alter application** **alter database**	*Oracle Essbase Technical Reference*
ESSCMD	DELETEVARIABLE	*Oracle Essbase Technical Reference*

Updating Substitution Variables

You can modify or update existing substitution variables. Before updating a substitution variable, see "Rules for Setting Substitution Variable Names and Values" on page 121.

➤ To update a substitution variable, use a tool:

Tool	Topic	See
Administration Services	Managing Substitution Variables	*Oracle Essbase Administration Services Online Help*
MaxL	**alter system** **alter application** **alter database**	*Oracle Essbase Technical Reference*
ESSCMD	UPDATEVARIABLE	*Oracle Essbase Technical Reference*

Copying Substitution Variables

You can copy substitution variables to any Essbase Server, application, or database to which you have appropriate access.

➤ To copy a substitution variable, see “Copying Substitution Variables” in the *Oracle Essbase Administration Services Online Help*.

Using Location Aliases

A location alias is a descriptor for a data source. A location alias maps an alias name for a database to the location of that database. A location alias is set at the database level and specifies an alias, a server, an application, a database, a user name, and a password. Set the location alias on the database on which the calculation script is run.

After you create a location alias, you can use the alias to refer to that database. If the location of the database changes, edit the location definition accordingly.

Note:

You can use location aliases only with the @XREF function. With this function, you can retrieve a data value from another database to include in a calculation on the current database. In this case, the location alias points to the database from which the value is to be retrieved. See the *Oracle Essbase Technical Reference*.

Creating Location Aliases

You can create a location alias for a particular database.

➤ To create a location alias, use a tool:

Tool	Topic	Location
Administration Services	Creating Location Aliases	*Oracle Essbase Administration Services Online Help*
MaxL	**create location alias**	*Oracle Essbase Technical Reference*
ESSCMD	CREATELOCATION	*Oracle Essbase Technical Reference*

Editing or Deleting Location Aliases

You can edit or delete location aliases that you created.

➤ To edit or delete a location alias, use a tool:

Tool	Topic	Location
Administration Services	Editing or Deleting Location Aliases	*Oracle Essbase Administration Services Online Help*
MaxL	**display location alias** **drop location alias**	*Oracle Essbase Technical Reference*
ESSCMD	LISTLOCATIONS DELETELOCATION	*Oracle Essbase Technical Reference*

7 Creating and Changing Database Outlines

In This Chapter

Some information in this chapter applies only to block storage databases and is not relevant to aggregate storage databases.

Also see:

- Chapter 57, "Comparison of Aggregate and Block Storage"
- Chapter 58, "Aggregate Storage Applications, Databases, and Outlines"

Process for Creating Outlines

This section provides an overview of creating outlines using Outline Editor. For information about Outline Editor, see the *Oracle Essbase Administration Services Online Help*.

For basic information about outlines, see Chapter 3, "Understanding Multidimensional Databases."

➤ To create an outline:

1 **Create a database. The new database automatically contains a blank outline.**

  See "Creating Applications and Databases" on page 119.

2 **Open the outline.**

  See "Creating and Editing Outlines" on page 128.

3 **Add dimensions and members to the outline.**

  See "Adding Dimensions and Members to an Outline" on page 129.

4 **Set each dimension as dense or sparse.**

See "Setting Data Storage Properties" on page 130.

5 **Position dimensions and members in the outline.**

See "Positioning Dimensions and Members" on page 130.

6 **Set dimension and member properties.**

See Chapter 9, "Setting Dimension and Member Properties."

7 **If necessary, create attribute dimensions and associate them with the appropriate base dimensions.**

See Chapter 10, "Working with Attributes."

8 **Verify and save the outline.**

See Figure 37 and "Saving Outlines" on page 133.

All examples in this chapter are based on the Sample.Basic database shipped with Essbase.

Creating and Editing Outlines

The database outline defines the structure of the database. Outline Editor displays the dimension hierarchy of an outline visually.

When a database is created, Essbase creates an outline for that database automatically. The outline has the same name as the database (`dbname.otl`) and is stored in the database directory on Essbase Server. You can create content in the new outline in the following ways:

- In Outline Editor, open the empty outline created by default when you create a database and add content manually.

  See "Opening and Editing Outlines" in the *Oracle Essbase Administration Services Online Help*.
- Copy an existing outline to the current database and change the existing outline.
- Create content using data sources and rules files.

  See Chapter 16, "Understanding Data Loading and Dimension Building."

Caution!

If you open the same outline with two instances of the Administration Services Console using the same login ID, each save overwrites the changes of the other instance. Oracle does not recommend this practice, as it can be difficult to keep track of the changes that are saved or overwritten.

➤ To create an outline or open an existing outline, use a tool:

Tool	Topic	Location
Administration Services	Opening and Editing Outlines	*Oracle Essbase Administration Services Online Help*
MaxL	**create database**	*Oracle Essbase Technical Reference*
ESSCMD	CREATEDB	*Oracle Essbase Technical Reference*

➤ To copy an existing outline, use a tool:

Tool	Topic	Location
Administration Services	Copying Outlines	*Oracle Essbase Administration Services Online Help*
MaxL	**create database as**	*Oracle Essbase Technical Reference*
ESSCMD	COPYDB	*Oracle Essbase Technical Reference*

Locking and Unlocking Outlines

In Outline Editor, an outline is always locked when it is opened in edit mode. Essbase unlocks the outline when the outline is closed. When an outline is locked, Essbase does not allow other users to save over, rename, delete, or edit the outline. When you attempt to edit a locked outline, you are given an option to view the outline in Outline Viewer.

If you have Administrator permissions, you can unlock a locked outline. Before you forcefully unlock a locked outline, make sure that no one else is working with it.

Note:

Essbase uses a different process for locking and unlocking outlines than for other database artifacts. See “Locking and Unlocking Artifacts” on page 719.

➤ To unlock an outline, use a tool:

Tool	Topic	Location
Administration Services	Locking and Unlocking Outlines	*Oracle Essbase Administration Services Online Help*
MaxL	**create database as**	*Oracle Essbase Technical Reference*
ESSCMD	UNLOCKOBJECT	*Oracle Essbase Technical Reference*

Adding Dimensions and Members to an Outline

You can add dimensions and member hierarchies to an outline in several ways:

- Manually, using Outline Editor
- With a data source and rules file, using Data Prep Editor

Before naming dimensions and members, see "Naming Restrictions for Applications and Databases" on page 1059.

➤ To add dimensions and members to an outline using Outline Editor, see "Adding Dimensions to Outlines" and "Adding Members to Dimensions" in the *Oracle Essbase Administration Services Online Help*.

➤ To add dimensions and members to an outline using Data Prep Editor, see "Creating a Dimension Build Rules File" in the *Oracle Essbase Administration Services Online Help*.

➤ To add dimensions and members dynamically (using a rules file) from Outline Editor, see "Updating an Outline Dynamically Using a Rules File" in the *Oracle Essbase Administration Services Online Help*.

Setting Data Storage Properties

When you create dimensions and save an outline, Essbase automatically sets the new dimensions in the outline as sparse. You can change the dimension storage type according to the optimal configuration for the database.

Set as sparse any standard dimensions with which you plan to associate attribute dimensions. See "Selection of Dense and Sparse Dimensions" on page 67.

➤ To set data storage properties using Outline Editor, see "Setting Dimensions as Dense or Sparse" in the *Oracle Essbase Administration Services Online Help*.

Positioning Dimensions and Members

Dimensions are the highest level of organization in an outline. Dimensions contain members, which you can nest inside of other members in a hierarchy. See "Dimensions and Members" on page 62.

The following sections describe how to position dimensions and members in the outline.

Note:

The relative locations of dimensions in an outline can affect calculation and retrieval performance times. See "Designing an Outline to Optimize Performance" on page 95.

Moving Dimensions and Members

After you create dimensions and members, you can rearrange them within the outline. Before moving members and dimensions in an outline, consider the following information:

- The positions of dimensions and members in an outline can affect performance.

  See "Optimizing Outline Performance" on page 170.
- Moving dimensions and members can affect the performance of calculations and retrievals.

  See "Designing an Outline to Optimize Performance" on page 95.
- Moving members could move a shared member before the actual member in the outline (which is not recommend).
- If you add, delete, or move nonattribute dimensions or members, Essbase restructures the database, and you must recalculate the data.
- Position attribute dimensions at the end of the outline. Otherwise, during outline verification, Essbase prompts you to move them there.

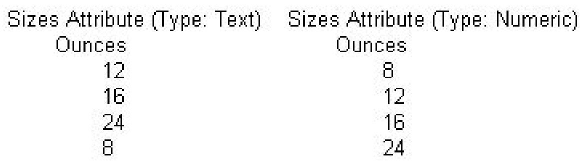
➤ To position dimensions and members using Outline Editor, see "Manipulating Dimensions and Members in an Outline" in the *Oracle Essbase Administration Services Online Help*.

Sorting Dimensions and Members

You can have Essbase arrange dimensions within an outline or members within a dimension in alphabetical order (A–Z) or reverse alphabetical order (Z–A). For a list of consequences of sorting dimensions and members, see "Moving Dimensions and Members" on page 131.

When you sort level 0 members of numeric attribute dimensions in outlines, the members are sorted by their values. For example, Figure 37 shows text and numeric versions of the Sizes attribute dimension after sorting the members in ascending order. The members of the numeric attribute dimension are sequenced by the numeric values of the members; the member 8 is before the other members. In the text attribute dimension, because the characters are sorted left to right, the member 8 is after the member 24.

Figure 37 Sorting Numeric Versus Text Attribute Dimension in Ascending Order



```
Sizes Attribute (Type: Text)   Sizes Attribute (Type: Numeric)
     Ounces                         Ounces
          12                             8
          16                             12
          24                             16
          8                              24
```

You cannot sort Boolean attribute dimensions. See "Understanding Attribute Types" on page 164.

➤ To sort members using Outline Editor, see “Sorting Members” in the *Oracle Essbase Administration Services Online Help*.

Verifying Outlines

You can verify an outline automatically when you save it, or you can verify the outline manually anytime. When verifying an outline, Essbase checks the following items:

- All member and alias names are valid. Members and aliases cannot have the same name as other members, aliases, generations, or levels. See “Naming Restrictions for Applications and Databases” on page 1059.
- Only one dimension is tagged as accounts, time, currency type, or country.
- Shared members are valid as described in “Understanding the Rules for Shared Members” on page 149.
- Level 0 members are not tagged as label only.
- Label-only members have not been assigned formulas.
- The currency category and currency name are valid for the currency outline.
- Dynamic Calc members in sparse dimensions do not have more than 100 children.
- If a parent member has one child, and if that child is a Dynamic Calc member, the parent member must also be Dynamic Calc.
- If a parent member has one child, and if that child is a Dynamic Calc, two-pass member, the parent member must also be Dynamic Calc, two-pass.
- The two names of members of Boolean attribute dimensions are the same as the two Boolean attribute dimension member names defined for the outline.
- The level 0 member name of a date attribute dimension must match the date format name setting (mm-dd-yyyy or dd-mm-yyyy). If the dimension has no members, because the dimension name is the level 0 member, the dimension name must match the setting.
- The level 0 member name of a numeric attribute dimension is a numeric value. If the dimension has no members, because the dimension name is the level 0 member, the dimension name must be a numeric value.
- Attribute dimensions are located at the end of the outline, following all standard dimensions.
- Level 0 Dynamic Calc members of standard dimensions have a formula.
- Formulas for members are valid.
- In a Hybrid Analysis outline, only the level 0 members of a dimension can be Hybrid Analysis-enabled.

During outline verify, Essbase also performs the following conversions to appropriate numeric attribute dimension member names and displays them in the outline:

- It moves minus signs in member names from the front to the end of the name; for example, –1 becomes 1–.

- It strips out leading or trailing zeroes in member names; for example, 1.0 becomes 1, and 00.1 becomes 0.1.

See "Understanding Attribute Types" on page 164.

➤ To verify an outline, see "Verifying Outlines" in the *Oracle Essbase Administration Services Online Help*.

Saving Outlines

You can save outlines to the Essbase Server or to a client computer or network. By default, Essbase saves outlines to the database directory on Essbase Server. If you are saving changes to an outline, Essbase may restructure the outline. For example, if you change a member name from Market to Region, Essbase moves data stored in reference to Market to Region. Each time that you save an outline, Essbase verifies the outline to ensure that it is correct.

➤ To save an outline, see "Saving Outlines" in the *Oracle Essbase Administration Services Online Help*.

Also see the following sections.

Saving an Outline with Added Standard Dimensions

If you add one or more new standard (nonattribute) dimensions, any data that existed previously in the database must be mapped to a member of each new dimension before the outline can be saved. For example, adding a dimension called Channel to the Sample.Basic outline implies that all previous data in Sample.Basic is associated with a particular channel or the sum of all channels.

Saving an Outline with One or More Deleted Standard Dimensions

If you delete one or more standard (nonattribute) dimensions, the data associated with only one member of each deleted dimension must be retained and associated with a member of one of the other dimensions. For example, removing a dimension called Market from the outline implies that all of the data that remains in the database after the restructure operation is associated with a single, specified member of the Market dimension.

If you delete an attribute dimension, Essbase deletes the associations to its base dimension. See Chapter 10, "Working with Attributes."

Creating Sub-Databases Using Deleted Members

➤ To create a sub-database:

1 **Delete a dimension from an existing outline.**

2 **Save the database using a different name, and specify the member to keep.**

Only one member can be kept when a dimension is deleted. See "Saving an Outline with One or More Deleted Standard Dimensions" on page 133.

8 Creating and Working With Duplicate Member Outlines

In This Chapter

The information in this chapter applies to block storage and aggregate storage databases.

Also see:

- Chapter 6, "Creating Applications and Databases"
- Chapter 7, "Creating and Changing Database Outlines"

Creating Duplicate Member Names in Outlines

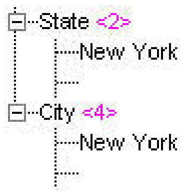
You can use duplicate names in an outline only if the outline that has the allow duplicate members option enabled.

When you enable duplicate member names in an outline, Essbase displays multiple members in the outline using the same name. Create the names in the usual way. See "Naming Restrictions for Dimensions, Members, and Aliases" on page 1060.

When you save the outline, Essbase validates and saves the outline with the duplicate member names. A qualified name format differentiates the duplicate member names.

Figure 38 shows an example of a duplicate member outline in which the New York state member and the New York city member appear in the outline as New York.

Figure 38 Duplicate Member Name "New York"



The qualified member names for the example in Figure 38 are [State].[New York] and [City].[New York]. See "Syntax for Specifying Duplicate Member Names and Aliases" on page 137.

➤ To create an outline that enables duplicate member names, or to convert a unique member name outline to a duplicate member name outline, use a tool:

Tool	Topic	Location
Administration Services	Creating Duplicate Member Name Outlines	*Oracle Essbase Administration Services Online Help*
MaxL	create database	*Oracle Essbase Technical Reference*

Note:

Save outline changes before converting it to a duplicate member name outline. You cannot convert an outline that has unsaved changes.

Note:

After converting an outline to a duplicate member outline, save it before proceeding with other outline changes.

Note:

A duplicate member outline cannot be converted back to a unique member outline.

Within a duplicate member outline, you can tag particular dimensions, generations, and levels as unique or duplicate to restrict the use of duplicate member names within a database. Doing so enables you to specify member name uniqueness at a more granular level in a duplicate member outline.

When you create a duplicate member outline, by default, all dimensions in the outline are tagged as duplicate.

When duplicate members are enabled in a dimension, you can tag particular generations or levels within the dimension as unique. If a member is assigned conflicting properties, the unique property takes precedence.

➤ To enable or disable duplicate member names in a dimension, see “Tagging a Dimension as Unique” in the *Oracle Essbase Administration Services Online Help*.

➤ To disable duplicate member names in a particular generation or level of a dimension, see “Naming Generations and Levels” in the *Oracle Essbase Administration Services Online Help*.

Note:

Duplicate member outline attribute dimensions do not have prefixes or suffixes attached that apply to attribute dimensions in unique outlines. For example, in a duplicate member Boolean attribute dimension, members do not include dimension, parent, grandparent, or ancestors affixed to the TRUE and FALSE members. See "Setting Prefix and Suffix Formats for Member Names of Attribute Dimensions" on page 171.

Restrictions for Duplicate Member Names and Aliases in Outlines

When creating duplicate member names and aliases in database outlines, the following must always be unique:

- Dimension names
- Generation names and level names
- Siblings under a parent member

If you are using aliases, this additional restriction applies: an alias table that contains duplicate alias names is valid only with a duplicate member outline.

Note:

Do *not* use quotation marks (" "), brackets ([ ]), or tabs in member, dimension, or alias names. For example, you cannot create a member name "[New York].[Area 1]". Outline verification does not display an error for member names that contain the invalid sequence of characters, and you can save the outline; however, Essbase cannot accurately query the data.

Syntax for Specifying Duplicate Member Names and Aliases

Although duplicate member names appear in the outline, each nonshared member name uniquely identifies a member in the database. A qualified name format differentiates the duplicate member names. When using Administration Services editors, you can select the qualified member name for an outline tree. You can view the qualified member name for a duplicate member in the Outline Viewer Member Properties dialog box in Administration Services. A qualified name must be used to specify a duplicate member name.

A qualified member or alias name can be specified in any of the following formats:

- Fully qualified member name
- Member name qualified by differentiating ancestor
- Shortcut qualified member name

Note:

A qualified name must comprise all alias names or all member names. You cannot mix member names and alias names in a qualified name.

Using Fully Qualified Member Names

A fully qualified member name comprises the duplicate member or alias name and *all* ancestors up to and including the dimension name. Each name must be enclosed in brackets ([ ]) and separated by a period (.). The syntax is as follows:

```
[DimensionMember].[Ancestors...].[DuplicateMember]
```

For example:

```
[Market].[East].[State].[New York]
[Market].[East].[City].[New York]
```

Qualifying Members by Differentiating Ancestor

A member name qualified by differentiating ancestor uses the member or alias name and *all* ancestors up to and including the ancestor that *uniquely* identifies the duplicate member or alias. The top ancestor in the path will always be a unique member name. Each name must be enclosed in brackets ([ ]) and separated by a period (.). The syntax is as follows:

```
[DifferentiatingAncestor].[Ancestors...].[DuplicateMember]
```

For example:

```
[State].[New York]
[City].[New York]
```

Using Shortcut Qualified Member Names

Essbase internally constructs shortcut qualified names for members in duplicate member outlines. These can be inserted into scripts using Administration Services by right-clicking the member and selecting Insert member name. You can also manually insert shortcut qualified names into scripts, spreadsheets, or MDX queries.

Essbase uses the syntax shown in Table 14 to construct shortcut qualified names. Using the same syntax that Essbase uses when you reference members in scripts, spreadsheets, and MDX queries is optimal but not required.

Table 14 Shortcut Qualified Name Syntax

Scenario	Qualified Name Syntax	Example

Duplicate member names exist at generation 2	`[DimensionMember].[DuplicateMember]`	`[Year].[Jan]`
		`[Product].[Jan]`

Duplicate member names exist in an outline but are unique within a dimension	`[DimensionMember]@[DuplicateMember]`	`[Year]@[Jan]`
Duplicate member names have a unique parent	`[ParentMember].[DuplicateMember]`	`[East].[New York]`

Scenario	Qualified Name Syntax	Example
Duplicate member names exist at generation 3	`[DimensionMember].[ParentMember].[DuplicateMember]`	`[Products].[Personal Electronics].[Televisions]`
Duplicate member names exist at a named generation or level, and the member is unique at its generation or level	`[DimensionMember]@[GenLevelName]|[DuplicateMember]`	`[2006]@[Gen1]|[Jan]`
In some scenarios, the differentiating ancestor method is used as a shortcut.	`[DifferentiatingAncestor].[Ancestors...].[DuplicateMember]`	`[2006].[Qtr1].[Jan]`

Working with Duplicate Member Names

This topic describes the syntax for defining duplicate member names in Administration Services.

To specify duplicate member names in:

- Smart View, see the *Oracle Hyperion Smart View for Office Online Help*
- Spreadsheet Add-in, see the *Oracle Essbase Spreadsheet Add-in Online Help*
- API, see the *Oracle Essbase API Reference*

To use duplicate member names in MaxL and MDX, see the *Oracle Essbase Technical Reference*.

Note:

If an alias name and member name are the same but do not represent the same member, searching by alias name is not supported in clients (for example Spreadsheet Add-in, Administration Services, or the API).

In Administration Services, if you use the member selection tool to insert a duplicate member name from the outline tree, the qualified member name is inserted automatically.

If you type a duplicate member name directly into an editor, type the qualified member name and enclose the qualified name in double quotation marks (" "). For example,

```
"[State].[New York]"
```

In MDX and MaxL qualified member names are *not* enclosed in quotation marks. See “Inserting Dimension and Member Names in MDX Scripts” in the *Oracle Essbase Administration Services Online Help*. The Administration Services Data Preview feature does not support duplicate member outlines.

9 Setting Dimension and Member Properties

In This Chapter

Some information in this chapter applies only to block storage databases and is not relevant to aggregate storage databases.

Also see:

- Chapter 57, "Comparison of Aggregate and Block Storage"
- Chapter 58, "Aggregate Storage Applications, Databases, and Outlines"

Setting 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 accounts, you can specify accounting measures for members in that dimension. Essbase calculates the two primary dimension types, time and accounts, before other dimensions in the database. By default, all dimensions are tagged as none.

The following sections describe the dimension types.

➤ To set a dimension type, see "Setting the Dimension Type" in the *Oracle Essbase Administration Services Online Help*.

Creating a Time Dimension

Tag a dimension as time if it contains members that describe how often you collect and update data. In the Sample.Basic database, for example, the Year dimension is tagged as time, as are its descendants—all Qtr members and the months (such as Jan). The time dimension also enables several accounts dimension functions, such as first and last time balances.

Rules when tagging a dimension as time:

- You can tag only one dimension in an outline as time.
- All members in the time dimension inherit the time property.
- You can add time members to dimensions that are not tagged as time.
- You can create an outline that does not have a time dimension.

➤ To tag a dimension as time, see "Tagging a Time Dimension" in the *Oracle Essbase Administration Services Online Help*.

Creating an Accounts Dimension

Tag a dimension as accounts if it contains items that you want to measure, such as profit or inventory.

Rules when tagging a dimension as accounts:

- You can tag only one dimension in an outline as accounts.
- All members in the accounts dimension inherit the accounts property.
- You can specify that members of the accounts dimension are calculated on the second pass through an outline. See "Setting Two-Pass Calculations" on page 156.
- You can create an outline that does not have an accounts dimension.

➤ To tag a dimension as accounts, see "Tagging an Accounts Dimension" in the *Oracle Essbase Administration Services Online Help*.

The following sections describe built-in functionality for accounts dimensions.

Setting Time Balance Properties

If an accounts dimension member uses the time balance property, it affects how Essbase calculates the parent of that member in the time dimension. By default, a parent in the time dimension is calculated based on the consolidation and formulas of its children. For example, in the Sample.Basic database, the Qtr1 member is the sum of its children (Jan, Feb, and Mar). However, setting a time balance property causes parents, for example Qtr1, to roll up differently.

➤ To set time balance properties, see “Setting Time Balance Properties” in the *Oracle Essbase Administration Services Online Help*.

Example of Time Balance as None

”None” is 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 the parent is based on the formulas and consolidation properties of its children.

Example of Time Balance as 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).

For example, assume that a member named OpeningInventory represents the inventory at the beginning of the time period. If the time period was Qtr1, OpeningInventory represents the inventory at the beginning of Jan; that is, the OpeningInventory for Qtr1 is the same as the OpeningInventory for Jan. For example, if you had 50 cases of Cola at the beginning of Jan, you also had 50 cases of Cola at the beginning of Qtr1.

Tag OpeningInventory as first, as shown in the following example consolidation:

```
OpeningInventory (TB First), Cola, East, Actual, Jan(+),  50
OpeningInventory (TB First), Cola, East, Actual, Feb(+),  60
OpeningInventory (TB First), Cola, East, Actual, Mar(+),  70
OpeningInventory (TB First), Cola, East, Actual, Qtr1(+), 50
```

Example of Time Balance as 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).

For example, assume that a member named EndingInventory represents the inventory at the end of the time period. If the time period is Qtr1, EndingInventory represents the inventory at the end of Mar; that is, the EndingInventory for Qtr1 is the same as the EndingInventory for Mar. For example, if you had 70 cases of Cola at the end of Mar, you also had 70 cases of Cola at the end of Qtr1.

Tag EndingInventory as last, as shown in the following example consolidation:

```
EndingInventory (TB Last), Cola, East, Actual, Jan(+),  50
EndingInventory (TB Last), Cola, East, Actual, Feb(+),  60
EndingInventory (TB Last), Cola, East, Actual, Mar(+),  70
EndingInventory (TB Last), Cola, East, Actual, Qtr1(+), 70
```

Example of Time Balance as Average

Set the time balance as “average” when you want the parent value to represent the average value of its children.

For example, assume that a member named AverageInventory represents the average of the inventory for the time period. If the time period was Qtr1, then AverageInventory represents the average of the inventory during Jan, Feb, and Mar.

Tag AverageInventory as average, as shown in the following example consolidation:

```
AverageInventory (TB Average), Cola, East, Actual, Jan(+),   60
AverageInventory (TB Average), Cola, East, Actual, Feb(+),   62
AverageInventory (TB Average), Cola, East, Actual, Mar(+),   67
AverageInventory (TB Average), Cola, East, Actual, Qtr1(+),  63
```

Setting Skip Properties

If you set the time balance as first, last, or average, set the skip property to tell Essbase what to do when it encounters missing values or values of 0.

Table 15 describes how each setting determines what Essbase does when it encounters a missing or zero value.

Table 15 Skip Properties

Setting	Essbase Action
None	Does not skip data when calculating the parent value.
Missing	Skips #MISSING data when calculating the parent value.
Zeros	Skips data that equals zero when calculating the parent value.
Missing and Zeros	Skips #MISSING data and data that equals zero when calculating the parent value.

If you mark a member as last with a skip property of missing or missing and zeros, the parent of that time period matches the last nonmissing child. In the following example, EndingInventory is based on the value for Feb, because Mar does not have a value.

```
Cola, East, Actual, Jan, EndingInventory (Last),  60
Cola, East, Actual, Feb, EndingInventory (Last),  70
Cola, East, Actual, Mar, EndingInventory (Last),  #MI
Cola, East, Actual, Qtr1, EndingInventory (Last), 70
```

Setting Variance Reporting Properties

Variance reporting properties determine how Essbase calculates the difference between actual and budget data in a member with the @VAR or @VARPER function in its member formula. Any member that represents an expense to the company requires an expense property.

When you are budgeting expenses for a time period, the actual expenses should be less than the budget. When actual expenses are greater than budget expenses, the variance is negative. The @VAR function calculates Budget – Actual. For example, if budgeted expenses are $100, and you spend $110, the variance is -10.

When you are budgeting nonexpense items, such as sales, the actual sales should be more than the budget. When actual sales are less than budget, the variance is negative. The @VAR function calculates Actual – Budget. For example, if budgeted sales were $100, and you made $110 in sales, the variance is 10.

By default, members are nonexpense.

➤ To set variance reporting properties, see "Setting Variance Reporting Properties" in the *Oracle Essbase Administration Services Online Help*.

Setting Essbase Currency Conversion Properties

Currency conversion properties define categories of currency exchange rates. These properties are used only in currency databases on members of accounts dimensions. See Chapter 13, "Designing and Building Currency Conversion Applications."

➤ To set currency conversion properties, see "Assigning Currency Categories to Accounts Members" in the *Oracle Essbase Administration Services Online Help*.

Creating a Country Dimension

Use country dimensions to track business activities in multiple countries. If you track business activity in the U.S. and Canada, for example, the country dimension should contain states, provinces, and countries. If a dimension is tagged as country, you can set the currency name property. The currency name property defines what type of currency this market region uses.

In a country dimension, you can specify the currency used in each member. For example, in the Interntl application and database shipped with Essbase, Canada has three markets—Vancouver, Toronto, and Montreal—which use Canadian dollars.

This dimension type is used for currency conversion applications. See Chapter 13, "Designing and Building Currency Conversion Applications."

➤ To tag a dimension as country, see "Tagging a Country Dimension" in the *Oracle Essbase Administration Services Online Help*.

Creating Currency Partitions

Use currency partition members to separate local currency members from a base currency defined in the application. If the base currency for analysis is U.S. dollars, for example, the local currency members would contain values based on the currency type of the region, such as Canadian dollars.

This dimension type is used for currency conversion applications. See Chapter 13, "Designing and Building Currency Conversion Applications."

➤ To tag a dimension as currency partition, see "Creating a Currency Partition" in the *Oracle Essbase Administration Services Online Help*.

Creating Attribute Dimensions

Use attribute dimensions to report and aggregate data based on characteristics of standard dimensions. In the Sample.Basic database, for example, the Product dimension is associated with the Ounces attribute dimension. Members of the Ounces attribute dimension categorize products based on their size in ounces.

Review the rules for using attribute dimensions in Chapter 10, “Working with Attributes.”

➤ To tag a dimension as an attribute, see “Tagging an Attribute Dimension” in the *Oracle Essbase Administration Services Online Help*.

Setting Member Consolidation

Member consolidation properties, which are listed in Table 16 on page 146, determine how children roll up into their parents. By default, new members are given the addition (+) operator, meaning that members are added. For example, Jan, Feb, and Mar figures are added and the result stored in their parent, Qtr1.

Note:

Essbase does not use consolidation properties with members of attribute dimensions. See “Calculating Attribute Data” on page 174.

Table 16 Consolidation Operators

Operator	Description
+	Adds the member to the result of previous calculations performed on other members. + is the default operator.
-	Multiplies the member by –1 and adds it to the sum of previous calculations performed on other members.
*	Multiplies the member by the result of previous calculations performed on other members.
/	Divides the member into the result of previous calculations performed on other members.
%	Divides the member into the sum of previous calculations performed on other members. The result is multiplied by 100 to yield a percentage value.
~	Does not use the member in the consolidation to its parent.
^	Does not use the member in any consolidation in any dimension.

➤ To set member consolidation properties, see “Setting Member Consolidation Properties” in the *Oracle Essbase Administration Services Online Help*.

Calculating Members with Different Operators

When siblings have different operators, Essbase calculates the data in top-down order. The following section describes how Essbase calculates the members listed below:

```
Parent1
```

```
   Member1  (+)    10
   Member2  (+)    20
   Member3  (-)    25
   Member4  (*)    40
   Member5  (%)    50
   Member6  (/)    60
   Member7  (~)    70
```

Essbase calculates Member1 through Member4 as follows:

```
(((Member1 + Member2) + (-1)Member3) * Member4) = X
(((10 + 20) + (-25)) * 40) = 200
```

If the result of this calculation is X, Member5 consolidates as follows:

```
(X/Member5) * 100 = Y
(200/50) * 100 = 400
```

If the result of the Member1 through Member4 calculation is Y, Member6 consolidates as follows:

```
Y/Member6 = Z
400/60 = 66.67
```

Because Member7 is set to No Consolidation(~), Essbase ignores Member7 in the consolidation.

Determining How Members Store Data Values

You can determine how and when Essbase stores the 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 discard the data value. Table 17 describes each storage property.

Table 17 Choosing Storage Properties

Storage Property	Behavior	For More Information
Store	Stores the data value with the member.	"Understanding Stored Members" on page 148
Dynamic Calc and Store	Does not calculate the data value until a user requests it, and then stores the data value.	"Understanding Dynamic Calculation Members" on page 148
Dynamic Calc	Does not calculate the data value until a user requests it, and then discards the data value.	"Understanding Dynamic Calculation Members" on page 148
Never share	Does not allow members to be shared implicitly. Members tagged as Never share can only be explicitly shared. To explicitly share a member, create the shared member with the same name and tag it as shared.	"Understanding Implied Sharing" on page 152
Label only	Creates members for navigation only; that is, members that contain no data values.	"Understanding Label Only Members" on page 148

Storage Property	Behavior	For More Information
Shared member	Shares values between members. For example, in the Sample.Basic database, the 100-20 member is stored under the 100 parent and shared under Diet parent.	"Understanding Shared Members" on page 149

➤ To set member storage properties, see "Setting Member Storage Properties" in the *Oracle Essbase Administration Services Online Help*.

Understanding Stored Members

Stored members contain calculated values that are stored with the member in the database after calculation. By default, members are set as stored.

➤ To define a member as stored, see "Setting Member Storage Properties" in the *Oracle Essbase Administration Services Online Help*.

Understanding Dynamic Calculation Members

When a member is Dynamic Calc, Essbase does not calculate the value for that member until a user requests it. After the user views it, Essbase does not store the value for that member. If you tag a member as Dynamic Calc and Store, Essbase performs the same operation as for a Dynamic Calc member but then stores the data value. See Chapter 26, "Dynamically Calculating Data Values."

Essbase automatically tags members of attribute dimensions as Dynamic Calc. You cannot change this setting.

➤ To tag a member as Dynamic Calc, see "Setting Member Storage Properties" in the *Oracle Essbase Administration Services Online Help*.

Understanding Label Only Members

Label only members have no associated data. Use them to group members or to ease navigation and reporting from the Spreadsheet Add-in. Typically, you should give label only members the "no consolidation" property. See "Setting Member Consolidation" on page 146.

You cannot associate attributes with label only members. If you tag as label only a base dimension member that has attribute associations, Essbase removes the attribute associations and displays a warning message.

➤ To tag a member as label only, see “Setting Member Storage Properties” in the *Oracle Essbase Administration Services Online Help*.

Understanding Shared Members

The data values associated with a shared member come from another member with the same name. The shared member stores a pointer to data contained in the other member, and the data is stored only once. To define a member as shared, an actual nonshared member of the same name must exist. For example, in the Sample.Basic database, the 100-20 member under 100 stores the data for that member. The 100-20 member under Diet points to that value.

Shared members typically are used to calculate the same member across multiple parents; for example, to calculate a Diet Cola member in both the 100 and Diet parents.

Using shared members lets you use members repeatedly throughout a dimension. Essbase stores the data value only once, but it displays in multiple locations. Storing the data value only once saves space and improves processing efficiency.

➤ To tag a member as shared, see “Setting Member Storage Properties” in the *Oracle Essbase Administration Services Online Help*.

Use the following sections to learn more about shared members.

Note:

Members with the same name may be duplicate members instead of shared members. See Chapter 8, “Creating and Working With Duplicate Member Outlines.”

Understanding the Rules for Shared Members

Rules when creating shared members:

- Shared members must be in the same dimension. For example, both 100-20 members in the Sample.Basic database are in the Product dimension.
- Shared members cannot have children.
- An unlimited number of shared members can have the same name.
- UDAs or formulas cannot be assigned to shared members.
- You can create a shared member for a member with a duplicate member name. Specify the duplicate member name for which you want to create the shared member. The qualified name of the duplicate member, on which the shared member is based, is displayed in the Member Properties dialog box. See “Defining Shared Members” in the *Oracle Essbase Administration Services Online Help*.
- Attributes cannot be associated with shared members.
- If accounts properties are assigned to shared members, the values for those accounts properties are taken from the base member, even if the accounts properties on the shared member are changed.

- Aliases can be assigned to shared members.
- An actual member must be located in a dimension before its shared member.
- Avoid complex relationships between actual and shared members that will be part of an attribute calculation, or a calculation may return unexpected results. See "Understanding Attribute Calculation and Shared Members" on page 178.

Note:

You cannot create a shared member and the member on which it is based under the same parent. In a duplicate member outline, siblings must be unique.

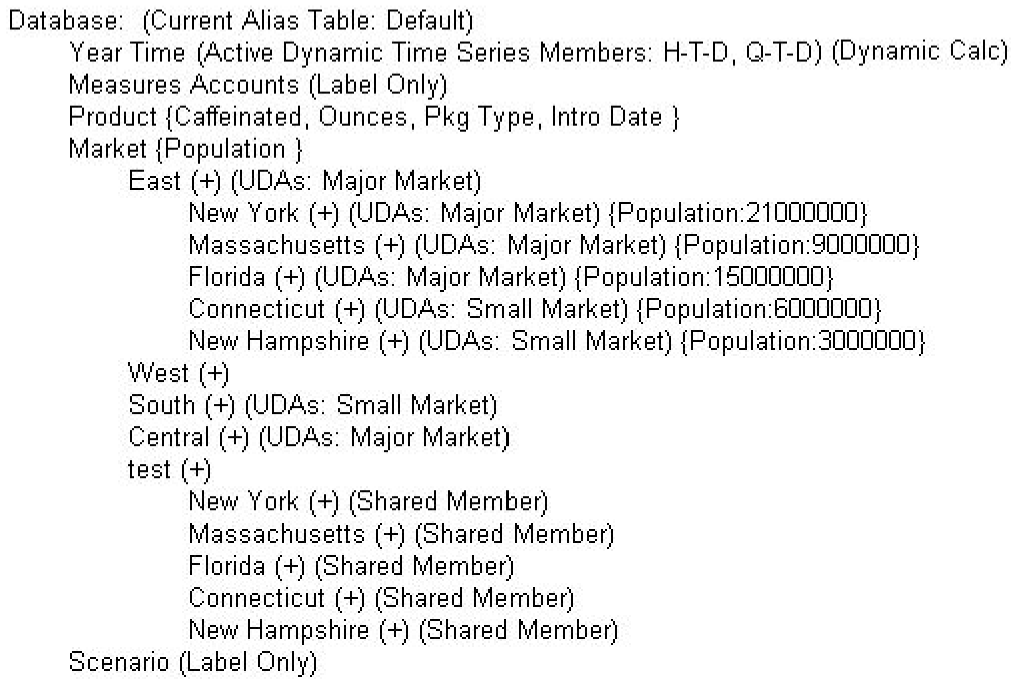
Understanding Shared Member Retrieval During Drill-Down

Essbase retrieves shared members during drill-down, depending on their location in the spreadsheet. Essbase follows three rules during this type of retrieval:

- Essbase retrieves stored members (not their shared member counterparts) by default.
- Essbase retrieves from the bottom of a spreadsheet first.
- If the parent of a shared member is a sibling of the stored member counterpart of one of the shared members, Essbase retrieves the stored member.

Example of Shared Members from a Single Dimension

If you create a test dimension with all shared members based on the members of the dimension East from the Sample.Basic outline, the outline would be similar to the following:



```
Database:  (Current Alias Table: Default)
    Year Time (Active Dynamic Time Series Members: H-T-D, Q-T-D) (Dynamic Calc)
    Measures Accounts (Label Only)
    Product {Caffeinated, Ounces, Pkg Type, Intro Date }
    Market {Population }
        East (+) (UDAs: Major Market)
            New York (+) (UDAs: Major Market) {Population:21000000}
            Massachusetts (+) (UDAs: Major Market) {Population:9000000}
            Florida (+) (UDAs: Major Market) {Population:15000000}
            Connecticut (+) (UDAs: Small Market) {Population:6000000}
            New Hampshire (+) (UDAs: Small Market) {Population:3000000}
        West (+)
        South (+) (UDAs: Small Market)
        Central (+) (UDAs: Major Market)
        test (+)
            New York (+) (Shared Member)
            Massachusetts (+) (Shared Member)
            Florida (+) (Shared Member)
            Connecticut (+) (Shared Member)
            New Hampshire (+) (Shared Member)
    Scenario (Label Only)
```

If you retrieve only the children of East, all results are from stored members because Essbase retrieves stored members by default.

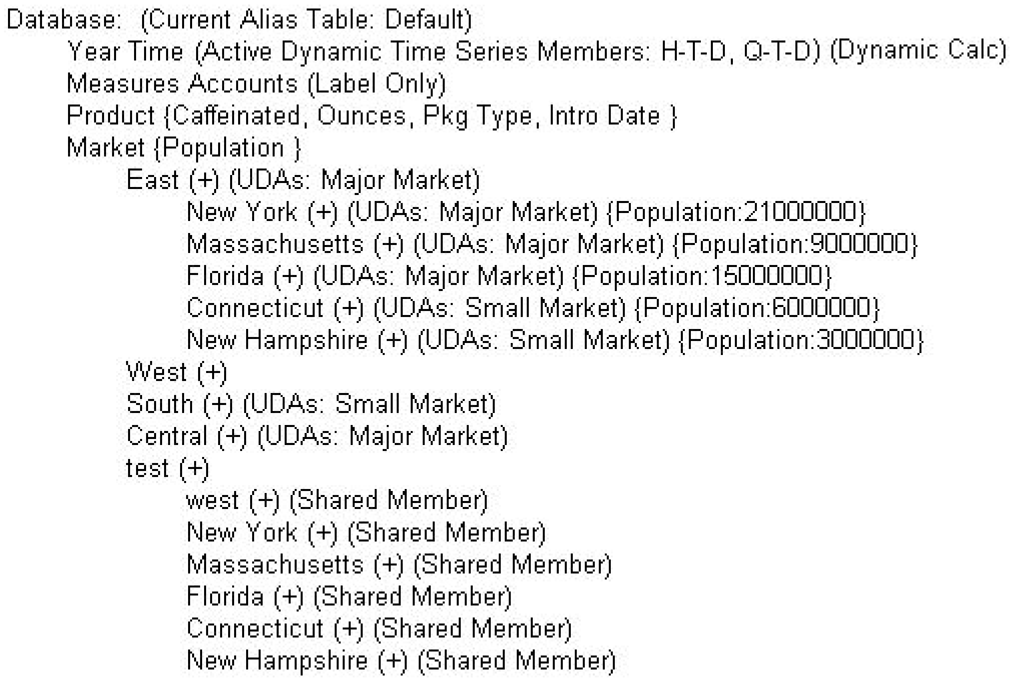
If, however, you retrieve data with the children of test above it in the spreadsheet, Essbase retrieves the shared members:

```
New York
Massachusetts
Florida
Connecticut
New Hampshire
test
```

If you move test above its last two children, Essbase retrieves the first three children as shared members, but the last two as stored members. Similarly, if you insert a member in the middle of the list above which was not a sibling of the shared members (for example, California inserted between Florida and Connecticut), Essbase retrieves shared members only between the nonsibling and the parent (in this case, between California and test).

Example of Retrieval with Crossed Generation Shared Members

You can modify the Sample.Basic outline to create a shared member whose stored member counterpart is a sibling to its own parent:



```
Database:  (Current Alias Table: Default)
    Year Time (Active Dynamic Time Series Members: H-T-D, Q-T-D) (Dynamic Calc)
    Measures Accounts (Label Only)
    Product {Caffeinated, Ounces, Pkg Type, Intro Date }
    Market {Population }
        East (+) (UDAs: Major Market)
            New York (+) (UDAs: Major Market) {Population:21000000}
            Massachusetts (+) (UDAs: Major Market) {Population:9000000}
            Florida (+) (UDAs: Major Market) {Population:15000000}
            Connecticut (+) (UDAs: Small Market) {Population:6000000}
            New Hampshire (+) (UDAs: Small Market) {Population:3000000}
        West (+)
        South (+) (UDAs: Small Market)
        Central (+) (UDAs: Major Market)
        test (+)
            west (+) (Shared Member)
            New York (+) (Shared Member)
            Massachusetts (+) (Shared Member)
            Florida (+) (Shared Member)
            Connecticut (+) (Shared Member)
            New Hampshire (+) (Shared Member)
```

If you create a spreadsheet with shared members in this order, Essbase retrieves all the shared members, except it retrieves the stored member West, not the shared member west:

```
West
New York
Massachusetts
Connecticut
New Hampshire
test
```

Essbase retrieves the members in this order because test is a parent of west and a sibling of west's stored member counterpart, West.

Understanding Implied Sharing

The shared member property defines a shared data relationship explicitly. Some members are shared even if you do not explicitly set them as shared. These members are *implied* shared members.

Essbase assumes (or implies) a shared member relationship in the following situations:

- **A parent has only one child.** In this situation, the parent and the child contain the same data. Essbase ignores the consolidation property on the child and stores the data only once —thus the parent has an implied shared relationship with the child. In the following example, the parent 500 has only one child, 500-10, so the parent shares the value of that child.

```
500 (+)
    500-10 (+)
```

- **A parent has only one child that consolidates to the parent.** If the parent has four children, but three are marked as no consolidation, the parent and child that consolidates contain the same data. Essbase ignores the consolidation property on the child and stores the data only once—thus the parent has an implied shared relationship with the child. In the following example, the parent 500 has only one child, 500-10, that rolls up to it. The other children are marked as No Consolidate(~), so the parent implicitly shares the value of 500-10.

```
500 (+)
    500-10 (+)
    500-20 (~)
    500-30 (~)
```

If you do not want a member to be shared implicitly, mark the parent as Never Share so that the data is duplicated instead. See "Understanding Shared Members" on page 149 for an explanation of how shared members work.

Setting Aliases

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 outlines or reports.

You can set more than one alias for a member using alias tables. 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 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

Essbase does not support aliases for Hybrid Analysis-enabled members.

Alias Tables

Aliases are stored in one or more tables as part of a database outline. An alias table maps a specific, named set of alias names to member names. When you create a database outline, Essbase creates an empty alias table named Default. If you do not create any other alias tables, the aliases that you create are stored in the Default alias table.

You can create an alias table for each set of outline members. When you view the outline or retrieve data, you can use the alias table name to indicate which set of alias names you want to see. Identifying which alias table contains the names that you want to see while viewing an outline is called making an alias table the active alias table. See "Setting an Alias Table as Active" on page 154.

For Unicode-mode applications, setting up a separate alias table for each user language enables users to view member names in their own language. See Chapter 41, "Understanding the Essbase Unicode Implementation."

Creating Aliases

You can provide an alias for any member. Alias names must follow the same rules as member names. See "Naming Restrictions for Dimensions, Members, and Aliases" on page 1060.

You can use any of the following methods to create aliases in an existing alias table:

➤ To manually assign an alias to a member while editing an outline, see "Creating Aliases for Dimensions and Members" in the *Oracle Essbase Administration Services Online Help*.

➤ To use dimension build and a data source to add aliases to an alias table, see "Defining a Rules File for Adding Aliases" in the *Oracle Essbase Administration Services Online Help*.

➤ To import alias values from an alias table source file created in a predefined format, see "Importing and Exporting Alias Tables" on page 155.

Creating and Managing Alias Tables

Named alias tables enable you to display different aliases in different situations. See "Alias Tables" on page 153. While working with alias tables, you can perform the actions described in the following sections.

Creating an Alias Table

An alias table contains a list of aliases to use for outline members. These restrictions apply:

- You can create up to 10 alias tables for an outline.
- The naming conventions for alias table names are the same as those for dimensions.

  See "Naming Restrictions for Dimensions, Members, and Aliases" on page 1060.

- Name-length restrictions depend on the Unicode-related mode of the application.

  See Appendix A, “Limits.”.
- Once an alias table is created, you cannot change its name.

➤ To create an alias table, see “Creating Alias Tables” in the *Oracle Essbase Administration Services Online Help*.

A new alias table is empty. To add aliases to an alias table and assign them to members, see “Creating Aliases” on page 153.

Setting an Alias Table as Active

The active alias table contains the aliases that Essbase currently displays in the outline.

➤ To view a list of alias tables in the outline and to set the current alias table, use a tool:

Tool	Topic	Location
Administration Services	Setting the Active Alias Table for Outline Editor	*Oracle Essbase Administration Services Online Help*
MaxL	**query database** **alter database**	*Oracle Essbase Technical Reference*
ESSCMD	LISTALIASES SETALIAS	*Oracle Essbase Technical Reference*

Copying an Alias Table

To copy an alias table, the table must be persisted in the database directory. To copy artifacts that are not persisted in the database directory, use the EXPORT ESSCMD command.

➤ To copy alias tables, use a tool:

Tool	Topic	Location
Administration Services	Copying Alias Tables	*Oracle Essbase Administration Services Online Help*
MaxL	**alter object**	*Oracle Essbase Technical Reference*
ESSCMD	COPYOBJECT	*Oracle Essbase Technical Reference*

Renaming an Alias Table

➤ To rename an alias table, use a tool:

Tool	Topic	Location
Administration Services	Renaming Alias Tables	*Oracle Essbase Administration Services Online Help*
MaxL	**alter object**	*Oracle Essbase Technical Reference*
ESSCMD	RENAMEOBJECT	*Oracle Essbase Technical Reference*

Clearing and Deleting Alias Tables

You can delete an alias table from the outline, or you can clear all the aliases from an alias table without deleting the alias table itself. To clear or delete alias tables, see "Deleting and Clearing Alias Tables" in the *Oracle Essbase Administration Services Online Help*.

Importing and Exporting Alias Tables

You can import a correctly formatted text file into Essbase as an alias table. Alias table import files have the `.alt` extension. Alias table import files should have the following structure:

- The first line in the file starts with $ALT_NAME. Add one or two spaces followed by the name of the alias table. If the alias table name contains a blank character, enclose the name in single quotation marks.
- The last line of the file must be $END.
- Each line between the first and the last lines contains two values separated by one or more spaces or tabs. The first value must be the name of an existing outline member; the second value is the alias for the member.
- Any member or alias name that contains a blank or underscore must be enclosed in double quotation marks.

The following is an example of an alias table import file:

```
$ALT_NAME  'Quarters'
Qtr1  Quarter1
Jan   January
Feb   February
Mar   March
$END
```

You can also export an alias table from the Essbase outline to a text file. The export file contains aliases and the corresponding member names—qualified member names for duplicate members.

➤ To import or export alias tables, use a tool:

Tool	Topic	Location
Administration Services	Importing Alias Tables Exporting Alias Tables	*Oracle Essbase Administration Services Online Help*
MaxL	**alter database**	*Oracle Essbase Technical Reference*
ESSCMD	LOADALIAS UNLOADALIAS	*Oracle Essbase Technical Reference*

Setting Two-Pass Calculations

By default, Essbase calculates outlines from the bottom up—first calculating the values for the children and then the values for the parent. Sometimes, however, the values of the children may be based on the values of the parent or the values of other members in the outline. To obtain the correct values for these members, Essbase must first calculate the outline and then recalculate the members that are dependent on the calculated values of other members. The members that are calculated on the second pass through the outline are called *two-pass calculations*.

See "Using Bottom-Up Calculation" on page 879.

For example, to calculate the ratio between Sales and Margin, Essbase needs first to calculate Margin, which is a parent member based on its children, including Sales. To ensure that the ratio is calculated based on a freshly calculated Margin figure, tag the Margin % ratio member as a two-pass calculation. Essbase calculates the database once and then calculates the ratio member again. This calculation produces the correct result.

Although two-pass calculation is a property that you can give to any nonattribute member, it works only on the following members:

- Accounts dimension members
- Dynamic Calc members
- Dynamic Calc and Store members.

If two-pass calculation is assigned to other members, Essbase ignores it.

➤ To tag a member as two-pass, see "Setting Two-Pass Calculation Properties" in the *Oracle Essbase Administration Services Online Help*.

Creating Formulas

You can apply formulas to standard dimensions and members. You cannot set formulas for attribute dimensions and their members. The formula determines how Essbase calculates the outline data. See Chapter 22, "Developing Formulas."

➤ To add formulas to a dimension or member, see “Creating and Editing Formulas in Outlines” in the *Oracle Essbase Administration Services Online Help*.

Naming Generations and Levels

You can create names for generations and levels in an outline, such as 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. See “Dimension and Member Relationships” on page 63.

Use generation and level names in calculation scripts or report scripts wherever you need to specify either a list of member names or generation or level numbers. For example, you could limit a calculation in a calculation script to all members in a specific generation. See Chapter 28, “Developing Calculation Scripts”.

You can define only one name for each generation or level. When you name generations and levels, follow the same naming rules as for members. See “Naming Restrictions for Dimensions, Members, and Aliases” on page 1060.

➤ To name generations and levels using Outline Editor, see “Naming Generations and Levels” in the *Oracle Essbase Administration Services Online Help*.

Creating UDAs

You can create user-defined attributes (UDA) for members. For example, you might create a UDA called Debit. Use UDAs in the following places:

- Calculation scripts. After you define a UDA, you can query a member for its UDA in a calculation script. For example, you can multiply all members with the UDA Debit by –1 so that they display as either positive or negative (depending on how the data is currently stored). See Chapter 28, “Developing Calculation Scripts.”
- Data loading. You can change the sign of the data as it is loaded into the database based on its UDA. See “Flipping Field Signs” on page 295.

To perform a calculation, selectively retrieve data based on attribute values, or provide full crosstab, pivot, and drill-down support in the spreadsheet, create attribute dimensions instead of UDAs. See “Comparing Attributes and UDAs” on page 166.

Rules when creating UDAs:

- You can define multiple UDAs per member.
- You cannot set the same UDA twice for one member.
- You can set the same UDA for different members.
- A UDA name can be the same as a member, alias, level, or generation name. Follow the same naming rules as for members. See “Naming Restrictions for Dimensions, Members, and Aliases” on page 1060.
- You cannot create a UDA on shared members.

- You cannot create a UDA on members of attribute dimensions.
- A UDA applies to the specified member only. Descendants and ancestors of the member do not automatically receive the same UDA.

➤ To add UDAs to a member, see "Working with UDAs" in the *Oracle Essbase Administration Services Online Help*.

Adding Comments

You can add comments to dimensions and members. Comments can contain 255 characters maximum. Outline Editor displays comments to the right of the dimension or member in the following format:

```
/* comment */
```

➤ To add comments to a dimension or member, see "Setting Comments on Dimensions and Members" in the *Oracle Essbase Administration Services Online Help*.

10 Working with Attributes

In This Chapter

Some information in this chapter applies only to block storage databases and is not relevant to aggregate storage databases.

Also see:

- Chapter 57, "Comparison of Aggregate and Block Storage"
- Chapter 58, "Aggregate Storage Applications, Databases, and Outlines"

For other information about using attributes, see:

- "Building Attribute Dimensions and Associating Attributes" on page 319
- Chapter 14, "Designing Partitioned Applications"
- Chapter 15, "Creating and Maintaining Partitions"
- Chapter 33, "Developing Report Scripts"

Process for Creating Attributes

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. This chapter describes how to create and manage attributes in an Essbase Server outline.

When working with attributes in Outline Editor, perform the following tasks:

1. Create a dimension.

   See "Adding Dimensions and Members to an Outline" on page 129. In the outline, position the attribute dimensions after all standard dimensions.

2. Tag the dimension as an attribute dimension and set attribute dimension type as text, numeric, Boolean, or date.

   See "Creating Attribute Dimensions" on page 145.
3. Add members to the attribute dimension.

   See "Adding Dimensions and Members to an Outline" on page 129.
4. Associate a base dimension with the attribute dimension.

   See "Understanding the Rules for Attribute Dimension Association" on page 163.
5. Associate members of the base dimension with members of the attribute dimension.

   See "Understanding the Rules for Attribute Member Association" on page 163.
6. If necessary, set up the attribute calculations.

   See "Calculating Attribute Data" on page 174.

Understanding Attributes

You can use the Essbase attribute feature to retrieve and analyze data not only from the perspective of dimensions, but also in terms of characteristics, or attributes, of those dimensions. For example, you can analyze product profitability based on size or packaging, and you can make more effective conclusions by incorporating into the analysis market attributes such as the population of each market region.

Such an analysis could tell you that decaffeinated drinks sold in cans in small markets (populations less than 6,000,000) are less profitable than you anticipated. For more details, you can filter the analysis by specific attribute criteria, including minimum or maximum sales and profits of different products in similar market segments.

A few ways analysis by attribute provides depth and perspective, supporting 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, and NOT operators and <, >, 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.
- Through the Attribute Calculations dimension automatically created by Essbase, you can view sums, counts, minimum or maximum values, and average values of attribute data. For example, when you enter Avg and Bottle into a spreadsheet, Essbase retrieves calculated values for average sales in bottles for all the column and row intersections on the sheet.
- You can perform calculations using numeric attribute values in calculation scripts and member formulas; for example, to determine profitability by ounce for products sized by the ounce.

- You can create crosstabs of attribute data for the same dimension, and you can pivot and drill down for detail data in spreadsheets.

  An attribute *crosstab* is a report or spreadsheet showing data consolidations across attributes of the same dimension. The crosstab example below displays product packaging as columns and the product size in ounces as rows. At their intersections, you see the profit for each combination of package type and size.

  From this information, you can see which size-packaging combinations were most profitable in the Florida market.

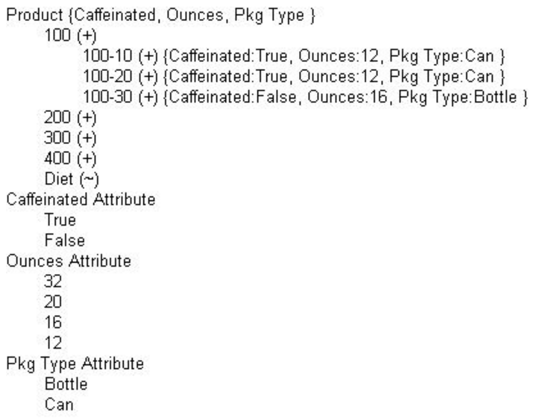
```
                    Product Year Florida Profit Actual

                           Bottle        Can           Pkg Type
                           =========     =========     =========
         32                  946           N/A            946
         20                  791           N/A            791
         16                  714           N/A            714
         12                  241         2,383          2,624
     Ounces                2,692         2,383          5,075
```

Understanding Attribute Dimensions

In the Sample.Basic database, products have attributes that are characteristics of the products. For example, products have an attribute that describes their packaging. In the outline, you see these characteristics as two dimensions, the Products dimension, and the Pkg Type attribute dimension that is associated with it. An *attribute* dimension has the word Attribute next to its name in the outline.

Figure 39 shows part of the Sample.Basic outline featuring the Product dimension and three attribute dimensions, Caffeinated, Ounces, and Pkg Type.

Figure 39 Outline Showing Base and Attribute Dimensions



```
Product {Caffeinated, Ounces, Pkg Type }
    100 (+)
        100-10 (+) {Caffeinated:True, Ounces:12, Pkg Type:Can }
        100-20 (+) {Caffeinated:True, Ounces:12, Pkg Type:Can }
        100-30 (+) {Caffeinated:False, Ounces:16, Pkg Type:Bottle }
    200 (+)
    300 (+)
    400 (+)
    Diet (~)
Caffeinated Attribute
    True
    False
Ounces Attribute
    32
    20
    16
    12
Pkg Type Attribute
    Bottle
    Can
```

In the outline, to the right of the Product dimension, the terms Caffeinated, Ounces, and Pkg Type show that these attribute dimensions are associated with the Product dimension.

A *standard* dimension is any dimension that is not an attribute dimension. When an attribute dimension is associated with a standard dimension, the standard dimension is the *base* dimension for that attribute dimension. In the outline in Figure 39, the Product dimension is the base dimension for the Caffeinated, Ounces, and Pkg Type attribute dimensions.

Note:

Attribute dimensions and members are Dynamic Calc, so Essbase calculates attribute information at retrieval time. Attribute data is not stored in the database.

Understanding Members of Attribute Dimensions

Members of an attribute dimension are potential attributes of the members of the associated base dimension. After you associate a base dimension with an attribute dimension, you associate members of the base dimension with members of the associated attribute dimension. The Market dimension member Connecticut is associated with the 6000000 member of the Population attribute dimension. That makes 6000000 an attribute of Connecticut.

In the outline, the information next to a base dimension member shows the attributes of that member. Figure 39, for example, shows that product 100-10 has three attributes—it has caffeine, is sold in 12-ounce containers, and is sold in cans.

Understanding the Rules for Base and Attribute Dimensions and Members

Rules regarding members of attribute dimensions and their base dimensions.

- You can tag only sparse dimensions as attribute dimensions.
- Before you can save an outline to the server, each attribute dimension must be associated with a standard, sparse dimension as its base dimension.
- Attribute dimensions must be the last dimensions in the outline.
- Attribute dimensions have a type setting—text, numeric, Boolean, or date. Text is the default setting. Although assigned at the dimension level, the type applies only to the level 0 members of the dimension. See "Understanding Attribute Types" on page 164.
- If you remove the attribute tag from a dimension, Essbase removes prefixes or suffixes from its member names. Prefixes and suffixes are not visible in the outline. See "Setting Prefix and Suffix Formats for Member Names of Attribute Dimensions" on page 171.
- A base dimension member can have many attributes, but only one attribute from each attribute dimension.

  For example, product 100-10 can have size and packaging attributes, but only one size and only one type of packaging.
- You cannot associate an attribute with an implied shared member, the child of which is tagged as shared.
- Essbase does not support attributes for Hybrid Analysis-enabled members.

You can use attribute values in calculations in the following comparisons:

- > (greater than)
- >= (greater than or equal to)
- < (less than)
- <= (less than or equal to)
- = = (equal to)
- <> or != (not equal to)
- IN

Understanding the Rules for Attribute Dimension Association

When you associate an attribute dimension with a standard dimension, the standard dimension is the *base* dimension for that attribute dimension.

- An attribute dimension must be associated with a sparse standard dimension.
- A standard dimension can be a base dimension for more than one attribute dimension.
- An attribute dimension can be associated with only one base dimension.

  For example, you might have a Size attribute dimension with members Small, Medium, and Large. If you associate the Size attribute dimension with the Product dimension, you cannot also associate the Size attribute dimension with the Market dimension. Tracking size-related information for the Market dimension requires another attribute dimension with a different name; for example, MarketSize, with the MarketSize attribute dimension associated with the Market dimension.

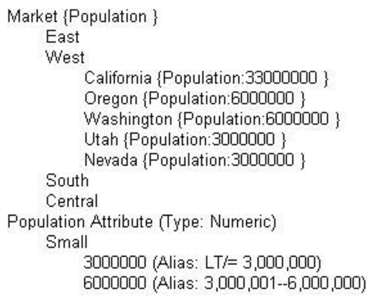
Understanding the Rules for Attribute Member Association

When you associate a member of an attribute dimension with a member of a base dimension, follow these rules:

- You cannot associate multiple members from the same attribute dimension with the same base dimension member. For example, the Bottle and Can package types cannot both be associated with the product 100-30.
- You can associate members from different attribute dimensions with the same member of a base dimension. For example, a decaffeinated cola product (100-30) sold in 16-ounce bottles has three attributes—Caffeinated:False; Ounces:16; and Pkg Type:Bottle.
- After attributes are associated with base dimension members, if you cut or copy and paste base dimension members to another outline location, the attribute associations are lost.
- Essbase does not require that each member of a base dimension be associated with a member of an attribute dimension.
- All base dimension members associated with members of a particular attribute dimension must be at the same level.

  For example, in Figure 40, all Market dimension members that have Population attributes are at level 0. You cannot associate East, which is a level 1 member, with a Population attribute, because the other members of the Market dimension that have Population attributes are level 0 members.

Figure 40 Association of Attributes with the Same Level Members of the Market Dimension



```
Market {Population }
    East
    West
        California {Population:33000000 }
        Oregon {Population:6000000 }
        Washington {Population:6000000 }
        Utah {Population:3000000 }
        Nevada {Population:3000000 }
    South
    Central
Population Attribute (Type: Numeric)
    Small
        3000000 (Alias: LT/= 3,000,000)
        6000000 (Alias: 3,000,001--6,000,000)
```

- The level 0 members of attribute dimensions are the only members that you can associate with base dimension members.

  For example, in the Population attribute dimension, you can associate only level 0 members such as 3000000, 6000000, and 9000000, with members of the Market dimension. You cannot associate a level 1 member such as Small.

  The name of the level 0 member of an attribute dimension is the attribute value. The only members of attribute dimensions that have attribute values are level 0 members.

  You can use the higher-level members of attribute dimensions to select and group data. For example, you can use Small, the level 1 member of the Population attribute dimension, to retrieve sales in both the 3000000 and 6000000 population categories.

Understanding Attribute Types

Attribute dimensions have a text, numeric, Boolean, or date type that enables different functions for grouping, selecting, or calculating data. Although assigned at the dimension level, the attribute type applies only to level 0 members of the attribute dimension.

- The default attribute type is text. Text attributes enable the basic attribute member selection and attribute comparisons in calculations. When you perform such comparisons, Essbase compares characters. For example, the package type Bottle is less than the package type Can, because B precedes C in the alphabet. In Sample.Basic, Pkg Type is an example of a text attribute dimension.
- The names of level 0 members of *numeric* attribute dimensions are numeric values. You can include the names (values) of numeric attribute dimension members in calculations. For example, you can use the number of ounces specified in the Ounces attribute to calculate profit per ounce for each product.

  You can also associate numeric attributes with ranges of base dimension values; for example, to analyze product sales by market population groupings—states with 3,000,000 population or less in one group, states with a population between 3,000,001 and 6 million in another group, and so on. See "Setting Up Member Names Representing Ranges of Values" on page 172.

- All Boolean attribute dimensions in a database contain only two members. The member names must match the settings for the database; for example, True and False. If more than one Boolean attribute dimension exists, specify a prefix or suffix member name format to ensure unique member names; for example, Caffeinated_True and Caffeinated_False. For a discussion of how to change Boolean names, see "Setting Boolean Attribute Member Names" on page 172.
- You can use date attributes to specify the date format—month-day-year or day-month-year—and to sequence information accordingly. For a discussion of how to change date formats, see "Changing the Member Names in Date Attribute Dimensions" on page 172. You can use date attributes in calculations. For example, you can compare dates in a calculation that selects product sales from markets established since 10-12-1999.

  Essbase supports date attributes from January 1, 1970, through January 1, 2038.

Comparing Attribute and Standard Dimensions

In general, attribute dimensions and their members are similar to standard dimensions and their members. You can provide aliases and member comments for attributes. Attribute dimensions can include hierarchies, and you can name generations and levels. You can perform the same spreadsheet operations on attribute dimensions and members as on standard dimensions and members; for example, to analyze data from different perspectives, you can retrieve, pivot, and drill down in the spreadsheet.

Table 18 describes major differences between attribute and standard dimensions and their members.

Table 18 Differences Between Attribute and Standard Dimensions

Functionality	Attribute Dimensions	Standard Dimensions
Storage	Sparse. Their base dimensions also must be sparse.	Can be dense or sparse
Storage property	Dynamic Calc only, therefore not stored in the database. The outline does not display this property.	Can be Store Data, Dynamic Calc and Store, Dynamic Calc, Never Share, or Label Only
Position in outline	Must be the last dimensions in the outline	Must be ahead of all attribute dimensions in the outline
Partitions	Cannot be defined along attribute dimensions, but you can use attributes to define a partition on a base dimension.	Can be defined along standard dimensions.
Formulas (on members)	Cannot be associated	Can be associated
Shared members	Not allowed	Allowed
Two-pass calculation member property	Not available	Available
Two-pass calculation with runtime formula	If a member formula contains a runtime-dependent function associated with an attribute member name, and the member with	Calculation is performed on standard members with runtime formulas and tagged two-pass.

Functionality	Attribute Dimensions	Standard Dimensions
	the formula is tagged as two-pass, calculation skips the member and issues a warning message. Runtime-dependent functions include: @CURRMBR, @PARENT, @PARENTVAL, @SPARENTVAL, @MDPARENTVAL, @ANCEST, @ANCESTVAL, @SANCESTVAL, and @MDANCESTVAL.	
Two-pass, multiple dimensions: Calculation order	Order of calculation of members tagged two-pass depends on order in outline. The last dimension is calculated last.	Calculation result is not dependent on outline order for members tagged two-pass in more than one dimension.
Two-pass calculation with no member formula	Calculation skipped, warning message issued. Therefore, member intersection of two-pass tagged members and upper-level members may return different results from calculation on standard dimensions.	Available
Dense Dynamic Calc members in nonexisting stored blocks	Calculations skip dense dimensions if they are on nonexisting stored blocks. To identify nonexisting stored blocks, export the database or run query to find out whether block has data.	Available
UDAs on members	Not allowed	Allowed
Consolidations	For all members, calculated through the Attribute Calculations dimension members: Sum, Count, Min, Max, and Avg. Consolidation operators in the outline are ignored during attribute calculations.	Consolidation operation indicated by assigning the desired consolidation symbol to each member
Member selection facilitated by Level 0 member typing	Available types include text, numeric, Boolean, and date.	All members treated as text
Associations	Must be associated with a base dimension	N/A
Spreadsheet drill-downs	List the base dimension data associated with the selected attribute. For example, drilling down on the attribute Glass displays sales for each product packaged in glass, where Product is the base dimension for the Pkg Type attribute dimension.	List lower or sibling levels of detail in the standard dimensions. For example, drilling down on QTR1 displays a list of products and their sales for that quarter.

Comparing Attributes and UDAs

Attributes and UDAs enable analysis based on characteristics of the data. Attributes provide greater capability than UDAs. The tables in this topic describe the differences between attributes and UDAs in these areas of functionality:

- Data storage (Table 19 on page 167)
- Data retrieval (Table 20 on page 167)
- Data conversion (Table 21 on page 168)
- Calculation scripts (Table 22 on page 168)

Table 19 Data Storage—Comparing Attributes and UDAs

Data storage	Attributes	UDAs
You can associate with sparse dimensions.	Supported	Supported
You can associate with dense dimensions.	Not supported	Supported

Table 20 Data Retrieval—Comparing Attributes and UDAs

Data Retrieval	Attributes	UDAs
You can group and retrieve consolidated totals by attribute or UDA value. For example, associate the value High Focus Item to various members of the Product dimension and use that term to retrieve totals and details for only those members.	Supported Simple	Supported More difficult to implement, requiring additional calculation scripts or commands
You can categorize attributes in a hierarchy and retrieve consolidated totals by higher levels in the attribute hierarchy; for example, if each product has a size attribute such as 8, 12, 16, or 32, and the sizes are categorized as small, medium, and large. You can view the total sales of small products.	Supported	Supported More difficult to implement
You can create crosstab views displaying aggregate totals of attributes associated with the same base dimension.	Supported You can show a crosstab of all values of each attribute dimension.	Supported You can retrieve only totals based on specific UDA values.
You can use Boolean operators AND, OR, and NOT with attribute and UDA values to refine a query. For example, you can select decaffeinated drinks from the 100 product group.	Supported	Supported
Because attributes have a text, Boolean, date, or numeric type, you can use appropriate operators and functions to work with and display attribute data. For example, you can view sales totals of all products introduced after a specific date.	Supported	Not supported
You can group numeric attributes into ranges of values and let the dimension building process automatically associate the base member with the appropriate range. For example, you can group sales in various regions based on ranges of their populations—less than 3 million, between 3 million and 6 million, and so on.	Supported	Not supported

Data Retrieval	Attributes	UDAs
Through the Attribute Calculations dimension, you can view aggregations of attribute values as sums, counts, minimums, maximums, and averages.	Supported	Not supported
You can use an attribute in a calculation that defines a member. For example, you can use the weight of a product in ounces to define the profit per ounce member of the Measures dimension.	Supported	Not supported
You can retrieve specific base members using attribute-related information.	Supported Powerful conditional and value-based selections	Supported Limited to text string matches only

Table 21 Data Conversion—Comparing Attributes and UDAs

Data Conversion	Attributes	UDAs
Based on the value of a UDA, you can change the sign of the data as it is loaded into the database. For example, you can reverse the sign of all members with the UDA Debit.	Not supported	Supported

Table 22 Calculation Scripts—Comparing Attributes and UDAs

Calculation Scripts	Attributes	UDAs
You can perform calculations on a member if its attribute or UDA value matches a specific value. For example, you can increase the price by 10% of all products with the attribute or UDA of Bottle.	Supported	Supported
You can perform calculations on base members whose attribute value satisfies conditions that you specify. For example, you can calculate the Profit per Ounce of each base member.	Supported	Not supported

Designing Attribute Dimensions

Essbase provides more than one way to design attribute information into a database. Most often, defining characteristics of the data through attribute dimensions and their members is the best approach. The following sections discuss when to use attribute dimensions, when to use other features, and how to optimize performance when using attributes.

Using Attribute Dimensions

For the most flexibility and functionality, use attribute dimensions to define attribute data. Using attribute dimensions provides the following features:

- Sophisticated, flexible data retrieval

You can view attribute data only when you want to; you can create meaningful summaries through crosstabs; and, using type-based comparisons, you can selectively view only the data that you want to see.

- Additional calculation functionality

  Not only can you perform calculations on the names of members of attribute dimensions to define members of standard dimensions, you can also access five types of consolidations of attribute data—sums, counts, averages, minimums, and maximums.

- Economy and simplicity

  Because attribute dimensions are sparse, Dynamic Calc, they are not stored as data. Compared to using shared members, outlines using attribute dimensions contain fewer members and are easier to read.

See "Understanding Attributes" on page 160.

Using Alternative Design Approaches

In some situations, consider one of the following approaches:

- UDAs. Although UDAs provide less flexibility than attributes, you can use them to group and retrieve data based on its characteristics. See "Comparing Attributes and UDAs" on page 166.
- Shared members. For example, to include a seasonal analysis in the Year dimension, repeat the months as shared members under the appropriate season; Winter: Jan (shared member), Feb (shared member), and so on. A major disadvantage of using shared members is that the outline becomes large if the categories repeat many members.
- Standard dimensions and members. Additional standard dimensions provide flexibility but add storage requirements and complexity to a database. For guidelines on evaluating the impact of additional dimensions, see "Analyzing and Planning" on page 81.

Table 23 describes situations in which you might consider an alternative approach to managing attribute data in a database.

Table 23 Considering Alternatives to Attribute Dimensions

Situation	Alternative
Analyze attributes of dense dimensions	UDAs or shared members
Perform batch calculation of data	Shared members or members of separate, standard dimensions
Define the name of a member of an attribute dimension as a value that results from a formula	Shared members or members of separate, standard dimensions
Define attributes that vary over time	Members of separate, standard dimensions. For example, to track product maintenance costs over time, the age of the product at the time of maintenance is important. However, using the attribute feature, you could associate only one age with the product. You need multiple members

Situation	Alternative
	in a separate dimension for each time period that you want to track.
Minimize retrieval time with large numbers of base-dimension members	Batch calculation with shared members or members of separate, standard dimensions.

Optimizing Outline Performance

Outline layout and content can affect attribute calculation and query performance. For general outline design guidelines, see "Designing an Outline to Optimize Performance" on page 95.

To optimize attribute query performance, consider the following design tips:

- Ensure that attribute dimensions are the only sparse Dynamic Calc dimensions in the outline.
- Locate sparse dimensions after dense dimensions in the outline. Place the most-queried dimensions at the beginning of the sparse dimensions and attribute dimensions at the end of the outline. In most situations, base dimensions are queried most.

See "Optimizing Calculation and Retrieval Performance" on page 177.

Building Attribute Dimensions

To build an attribute dimension, tag the dimension as an attribute and assign the dimension a type. Then associate the attribute dimension with a base dimension. Finally, associate each level 0 member of the attribute dimension with a member of the associated base dimension.

➤ To build an attribute dimension, see "Defining Attributes" in the *Oracle Essbase Administration Services Online Help*.

➤ To view the dimension, attribute value, and attribute type of a specific attribute member, use a tool:

Tool	Topic	Location
Administration Services	Viewing Attribute Information in Outlines	*Oracle Essbase Administration Services Online Help*
MaxL	**query database**	*Oracle Essbase Technical Reference*
ESSCMD	GETATTRINFO	*Oracle Essbase Technical Reference*

Setting Member Names in Attribute Dimensions

When you use the attribute feature, Essbase establishes default member names; for example, the system-defined True and False precludes other member names of True and False. You can change these system-defined names for the database. Date attributes and numeric attributes also can be duplicated. To avoid duplicate name confusion, you can establish settings for qualifying member names in attribute dimensions. The outline does not show the fully qualified attribute names, but you can see the full attribute names anywhere you select members, such as when you define partitions or select information to be retrieved.

Define the member name settings before you define or build the attribute dimensions. Changing the settings after the attribute dimensions and members are defined could result in invalid member names.

The following sections describe how to work with the names of members of attribute dimensions:

Note:

If you partition on outlines containing attribute dimensions, the name format settings of members described in this section must be identical in the source and target outlines.

Setting Prefix and Suffix Formats for Member Names of Attribute Dimensions

The information in this section does not apply to duplicate member attribute dimensions.

The names of members of Boolean, date, and numeric attribute dimensions are values. It is possible to encounter duplicate attribute values in different attribute dimensions.

- Boolean example

  If you have more than one Boolean attribute dimension in an outline, the two members of each of those dimensions have the same names, by default, True and False.

- Date example

  If you have more than one date attribute dimension, some member names in both dimensions could be the same. For example, the date on which a store opens in a certain market could be the same as the date on which a product was introduced.

- Numeric example

  The attribute value for the size of a product could be 12, and 12 also could be the value for the number of packing units for a product. This example results in two members with the name 12.

You can define unique names by attaching a prefix or suffix to member names in Boolean, date, and numeric attribute dimensions in the outline. You can affix the dimension, parent, grandparent, or all ancestors to the attribute name. For example, by setting member names of attribute dimensions to include the dimension name as the suffix, attached by an underscore, the member value 12 in the Ounces attribute dimension assumes the unique, full attribute member name 12_Ounces.

By default, Essbase assumes that no prefix or suffix is attached to the names of members of attribute dimensions.

The convention that you select applies to the level 0 member names of all numeric, Boolean, and date attribute dimensions in the outline. You can define aliases for these names if you want to display shorter names in retrievals.

➤ To define prefix and suffix formats, see “Defining a Prefix or Suffix Format for Members of Attribute Dimensions” in the *Oracle Essbase Administration Services Online Help*.

Setting Boolean Attribute Member Names

When you set the dimension type of an attribute dimension as Boolean, Essbase automatically creates two level 0 members with the names specified for the Boolean attribute settings. The initial Boolean member names in a database are set as True and False. To change these default names, for example, to Yes and No, define the member names for Boolean attribute dimensions before you create Boolean attribute dimensions in the database.

Before you can set an attribute dimension type as Boolean, you must delete all existing members in the dimension.

➤ To define the database setting for the names of members of Boolean attribute dimensions, see “Setting Member Names for Boolean Attribute Dimensions” in the *Oracle Essbase Administration Services Online Help*.

Changing the Member Names in Date Attribute Dimensions

You can change the format of members of date attribute dimensions. For example, you can use the following date formats:

- mm-dd-yyyy: October 18, 2007 is displayed as 10-18-2007.
- dd-mm-yyyy: October 18, 2007 is displayed as 18-10-2007.

If you change the date member name format, the names of existing members of date attribute dimensions may be invalid. For example, if the 10-18-2007 member exists, and you change the format to dd-mm-2007, outline verification will find this member invalid. If you change the date format, you must rebuild the date attribute dimensions.

➤ To change member names in date attribute dimensions, see “Setting the Member Name Format of Date Attribute Dimensions” in the *Oracle Essbase Administration Services Online Help*.

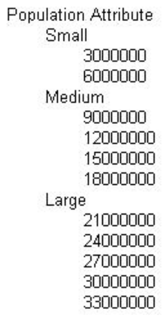
Setting Up Member Names Representing Ranges of Values

Members of numeric attribute dimensions can represent single numeric values or ranges of values:

- Single-value example: the member 12 in the Ounces attribute dimension represents the single numeric value 12; you associate this attribute with all 12-ounce products. The outline includes a separate member for each size; for example, 16, 20, and 32.

- Range of values example: the Population attribute dimension, as shown in Figure 41:

Figure 41 Population Attribute Dimension and Members



```
Population Attribute
    Small
        3000000
        6000000
    Medium
        9000000
        12000000
        15000000
        18000000
    Large
        21000000
        24000000
        27000000
        30000000
        33000000
```

  In this outline, the members of the Population attribute dimension represent ranges of population values in the associated Market dimension. The 3000000 member represents populations from zero through 3,000,000; the 6000000 member represents populations from 3,000,001 through 6,000,000; and so on. A setting for the outline establishes that each numeric member represents the top of its range.

  You can also define this outline setting so that members of numeric attribute dimensions are the bottoms of the ranges that they represent. For example, if numeric members are set to define the bottoms of the ranges, the 3000000 member represents populations from 3,000,000 through 5,999,999, and the 6000000 member represents populations from 6,000,000 through 8,999,999.

When you build the base dimension, Essbase automatically associates members of the base dimension with the appropriate attribute range. For example, if numeric members represent the tops of ranges, Essbase automatically associates the Connecticut market, with a population of 3,269,858, with the 6000000 member of the Population attribute dimension.

In the dimension build rules file, specify the size of the range for each member of the numeric attribute dimension. In the above example, each attribute represents a range of 3,000,000.

➤ To define ranges in numeric attribute dimensions, see “Assigning Member Names to Ranges of Values” in the *Oracle Essbase Administration Services Online Help*.

Note:

Oracle recommends that numeric attribute dimension member names contain no more than six decimal positions. Otherwise, because of precision adjustments, an outline may not pass verification.

Changing the Member Names of the Attribute Calculations Dimension

To avoid duplicating names in an outline, you may need to change the name of the Attribute Calculations dimension or its members. See “Understanding the Attribute Calculations Dimension” on page 174.

Regardless of the name that you use for a member, its function remains the same. For example, the second (Count) member always counts.

➤ To change member names in the Attribute Calculations dimension, see "Changing Member Names of Attribute Calculations Dimensions" in the *Oracle Essbase Administration Services Online Help*.

Calculating Attribute Data

Essbase calculates attribute data dynamically at retrieval time, using members from a system-defined dimension created by Essbase. Using this dimension, you can apply different calculation functions, such as a sum or an average, to the same attribute. You can also perform specific calculations on members of attribute dimensions; for example, to determine profitability by ounce for products sized by the ounce.

The following information assumes that you understand the concepts of attribute dimensions and Essbase calculations, including dynamic calculations. See the following sections.

Understanding the Attribute Calculations Dimension

When you create the first attribute dimension in the outline, Essbase also creates the Attribute Calculations dimension comprising five members with the default names `Sum`, `Count`, `Min` (minimum), `Max` (maximum), and `Avg` (average). You can use these members in spreadsheets or in reports to dynamically calculate and report on attribute data, such as the average yearly sales of 12-ounce bottles of cola in the West.

The Attribute Calculations dimension is not visible in the outline. You can see it wherever you select dimension members, such as in the Spreadsheet Add-in.

The attribute calculation dimension has the following properties:

- System-defined

  When you create the first attribute dimension in an application, Essbase creates the Attribute Calculations dimension and its members (Sum, Count, Min, Max, and Avg). Each member represents a type of calculation to be performed for attributes.

  See "Understanding the Default Attribute Calculations Members" on page 175.

- Label only

  Like all label only dimensions, the Attribute Calculations dimension shares the value of its first child, Sum.

  See "Member Storage Properties" on page 94.

- Dynamic Calc.

  The data in the Attribute Calculations dimension is calculated when a user requests it and is then discarded. You cannot store calculated attribute data in a database.

  See Chapter 26, "Dynamically Calculating Data Values."

- Not displayed in Outline Editor.

  The Attribute Calculations dimension is not displayed in Outline Editor. Members from this dimension can be viewed in spreadsheets and in reports.

There is no consolidation along attribute dimensions. You cannot tag members from attribute dimensions with consolidation symbols (for example, + or -) or with member formulas in order to calculate attribute data. As Dynamic Calc members, attribute calculations do not affect the batch calculation in terms of time or calculation order.

To calculate attribute data at retrieval time, Essbase performs the following tasks:

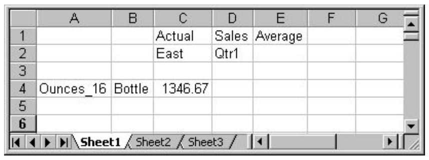
1. Finds the base-dimension members associated with the attribute-dimension members present in the current query
2. Dynamically calculates the sum, count, minimum, maximum, or average for the attribute-member combination for the current query
3. Displays the results in the spreadsheet or report
4. Discards the calculated values—that is, the values are not stored in the database

Note:

Essbase excludes #MISSING values when calculating attribute data.

For example, as shown in Figure 42, a spreadsheet user specifies two members of attribute dimensions (Ounces_16 and Bottle) and an Attribute Calculations member (Avg) in a spreadsheet report. Upon retrieval, Essbase dynamically calculates the average sales values of all products associated with these attributes for the current member combination (Actual -> Sales -> East -> Qtr1):

Figure 42 Retrieving an Attribute Calculations Member



	A	B	C	D	E	F	G
1			Actual	Sales	Average		
2			East	Qtr1			
3							
4	Ounces_16	Bottle	1346.67				
5							
6							

See "Accessing Attribute Calculations Members Using the Spreadsheet" on page 177.

Understanding the Default Attribute Calculations Members

The Attribute Calculations dimension contains five members used to calculate and report attribute data:

- Sum calculates a sum, or total, of the values for a member with an attribute or combination of attributes.
- Count calculates the number of members with the specified attribute or combination of attributes, for which a data value exists. Count includes only those members that have data blocks in existence. To calculate a count of all members with certain attributes, regardless

of whether they have data values, use the @COUNT function in combination with the @ATTRIBUTE function. See the *Oracle Essbase Technical Reference*.

- Avg calculates a mathematical mean, or average, of the nonmissing values for an specified attribute or combination of attributes (Sum divided by Count).
- Min calculates the minimum data value for a specified attribute or combination of attributes.
- Max calculates the maximum data value for a specified attribute or combination of attributes.

Note:

Each of these calculations excludes #MISSING values.

You can change these default member names, subject to the same naming conventions as standard members. For a discussion of Attribute Calculations member names, see "Changing the Member Names of the Attribute Calculations Dimension" on page 173.

Viewing an Attribute Calculation Example

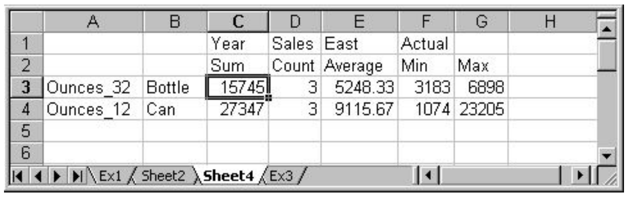
As an example of how Essbase calculates attribute data, consider the following yearly sales data for the East:

Table 24 Sample Attribute Data

Base-Dimension Member	Associated Attributes	Sales Value for Attribute-Member Combination
Cola	Ounces_12, Can	23205
Diet Cola	Ounces_12, Can	3068
Diet Cream	Ounces_12, Can	1074
Grape	Ounces_32, Bottle	6398
Orange	Ounces_32, Bottle	3183
Strawberry	Ounces_32, Bottle	5664

Figure 43 on page 176 shows how calculated attribute data might look in a spreadsheet report. You can retrieve multiple Attribute Calculations members for attributes. For example, you can calculate Sum, Count, Avg, Min, and Max for bottles and cans.

Figure 43 Sample Spreadsheet with Attribute Data



	A	B	C	D	E	F	G	H
1			Year	Sales	East	Actual		
2			Sum	Count	Average	Min	Max	
3	Ounces_32	Bottle	15745	3	5248.33	3183	6898	
4	Ounces_12	Can	27347	3	9115.67	1074	23205	
5								
6								

Accessing Attribute Calculations Members Using the Spreadsheet

You can access members from the Attribute Calculations dimension in Spreadsheet Add-in. From the spreadsheet, users can view Attribute Calculations dimension members using any of the following methods:

- Entering members directly into a sheet
- Selecting members from the Query Designer
- Entering members as an EssCell parameter

See the *Oracle Essbase Spreadsheet Add-in User's Guide*.

Optimizing Calculation and Retrieval Performance

To optimize attribute calculation and retrieval performance, consider the following:

- The calculation order for attribute calculations is the same as for dynamic calculations. For an outline, see "Calculation Order for Dynamic Calculation" on page 425.
- Because Essbase calculates attribute data dynamically at retrieval time, attribute calculations do not affect the performance of the overall (batch) database calculation.
- Tagging base-dimension members as Dynamic Calc may increase retrieval time.
- When a query includes the Sum member and an attribute-dimension member whose associated base member is tagged as two-pass, retrieval time may be slow.
- To maximize attribute retrieval performance, use any of the following techniques:
    - Configure the outline using the tips in "Optimizing Outline Performance" on page 170.
    - Drill down to the lowest level of base dimensions before retrieving data. For example, in Spreadsheet Add-in, turn on the Navigate Without Data feature, drill down to the lowest level of the base dimensions included in the report, and then retrieve data.
    - When the members of a base dimension are associated with several attribute dimensions, consider grouping the members of the base dimension according to their attributes. For example, in the Sample.Basic database, you can group all 8-ounce products.

Using Attributes in Calculation Formulas

In addition to using the Attribute Calculations dimension to calculate attribute data, you can use calculation formulas on members of standard or base dimensions to perform specific calculations on members of attribute dimensions; for example, to determine profitability by ounce for products sized by the ounce.

You cannot associate formulas with members of attribute dimensions.

Note:

Some restrictions apply when using attributes in formulas associated with two-pass members. See the rows about two-pass calculations in Table 18 on page 165.

Table 25lists functions you can use to perform specific calculations on attributes:

Table 25 Functions That Calculate On Attributes

Type of Calculation	Function to Use
Generate a list of all base members with a specific attribute. For example, generate a list of members that have the Bottle attribute, and then increase the price for those members.	@ATTRIBUTE
Return the value of the level 0 attribute member that is associated with the base member being calculated. ● From a numeric or date attribute dimension (using @ATTRIBUTEVAL) ● From a Boolean attribute dimension (using @ATTRIBUTEBVAL) ● From a text attribute dimension (using @ATTRIBUTESVAL) For example, return the numeric value of a size attribute (for example, 12 for the member 12 under Ounces) for the base member being calculated (for example, Cola).	@ATTRIBUTEVAL @ATTRIBUTEBVAL @ATTRIBUTESVAL
Convert a date string to numbers for a calculation. For example, use @TODATE in combination with the @ATTRIBUTEVAL function to increase overhead costs for stores opened after a certain date.	@TODATE
Generate a list of base dimension members associated with attributes that satisfy the conditions that you specify. For example, generate a list of products that are greater than or equal to 20 ounces, and then increase the price for those products.	@WITHATTR

Note:

For syntax information and examples for these functions, see the *Oracle Essbase Technical Reference*. For an additional example using @ATTRIBUTEVAL in a formula, see "Calculating an Attribute Formula" on page 382.

Understanding Attribute Calculation and Shared Members

Attribute calculations start at level 0 and stop at the first stored member. Therefore, if your outline has placed a stored member between two shared members in a an outline hierarchy, the calculation results may not include the higher shared member.

For example:

```
Member 1 (stored)
    Member A (stored)
        Member 2 (shared)
Member B (stored)
    Member 1 (shared member whose stored member is Member 1 above)
```

In this example, when an attribute calculation is performed, the calculation starts with level 0 Member 2 and stops when it encounters the first stored member, Member A. Therefore, Member 1 would not be included in the calculation.

To avoid unexpected results with attribute calculation, avoid mixing shared and stored members. For this example, if Member 2 were not shared, or Member 1 did not have a corresponding shared member elsewhere in the outline, calculation results would be as expected.

Varying Attributes

About Varying Attributes

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.

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.

Note:

There are two types of independent dimensions: continuous and discrete. The members in a continuous dimension reflect continuity. For example, week, month, and quarter reflect the continuity in a time dimension. The members in a discrete dimension do not imply continuity. For example, California, Texas, and Ohio in a market dimension do not have a relationship based on continuity.

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 26 Varying Attribute Example: Sales Representative 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 the use of 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, the retrieval results would be similar to the following table:

Table 27

SUBSKU	SKU
Red	100
White	400
White	600
Black	200
Black	300
Silver	500

Varying attributes must meet the following guidelines:

- They must have multiple chains.
- Leaf levels must match.

You can enable an outline to support varying attributes. You can define attribute dimensions to function as varying attributes. You can also edit varying attributes to reflect the type of information you need.

Implementing Varying Attributes

Varying attributes are supported for aggregate and block storage databases. You implement varying attributes at the database level.

Use the following workflow to enable and use varying attributes:

1. In the outline properties, enable varying attributes.
2. In the member properties for the base dimension, go to the Attributes tab and identify the independent dimension (the dimension upon which varying attributes depend).

For example, if the sales representative attribute association for Customer A gets changed in May, then Year would be the independent dimension.

3. Specify the type of independent dimension: continuous or discrete. An example of a continuous independent dimension is one that is based on time. A discrete independent dimension has no continuity; for example, in a Market dimension, California, Texas, and Ohio do not have a relationship based on continuity.
4. Associate the independent dimension with a varying attribute. Optionally select a *range* and an *association mode*.

   A range can be assigned for which the attribute association is true. For example, you can assign a time range for which the attribute association applies: Jane is an Engineer from July 2007–June 2008.

   An association mode tells Essbase how to handle conflicting associations of a varying attribute with its independent dimension. The following association modes are available: Overwrite, NoOverwrite, and Extend.
5. Save and restructure the outline.
6. Perform the following maintenance tasks as needed:
   - Add new varying attribute associations to independent members (for example, add a new job title for an employee).
   - Remove independent member associations.
   - View existing independent dimension member associations (for example, see for which Months the company had a replacement sales manager).
   - Disassociate attribute dimensions from base dimensions.

Functions Supporting Varying Attributes

The following Report Writer functions are designed to work with varying attributes.

- <PERSPECTIVE
- <WITHATTREX
- <ATTRIBUTEVA

For more information, see the *Oracle Essbase Technical Reference*.

The following MDX functions are designed to work with varying attributes.

- AttributeEx
- WithAttrEx
- The WITH PERSPECTIVE keywords

For more information, see the *Oracle Essbase Technical Reference*.

Limitations of Varying Attributes

Continuous independent dimensions must act as a single dimension, for example Year and Month. Unconnected continuous independent dimensions are not supported.

Continuous independent dimensions and members must be specified last.

Independent members must be stored, level 0 members.

11 Linking Objects to Essbase Data

In This Chapter

The information in this chapter applies only to block storage databases and is not relevant to aggregate storage databases. Also see Chapter 57, “Comparison of Aggregate and Block Storage.”

Understanding LROs

You can link various kinds of data with any cell in an Essbase database, using a linked reporting object (LRO). This ability is similar to attaching a file to e-mail. An LRO provides improved support for planning and reporting applications and can enhance your data analysis capabilities.

LROs are objects (alternatively called artifacts) that you associate with specific data cells in an Essbase database. Users create linked objects through Spreadsheet Add-in by selecting a data cell and choosing a menu item. You can link an unlimited number of objects to a cell. The objects are stored on the Essbase Server, where they are available to any user with the appropriate access permissions. Users retrieve and edit the objects through the Spreadsheet Add-in Linked Objects Browser feature, which enables them to view objects linked to the selected cell. For the maximum sizes of the types of linked objects described in Table 28, see Appendix A, “Limits.”

Table 28 Types of Linked Objects

Object Type	Description
Cell note	A text annotation
File	An external file, such as a Microsoft Word document, an Excel spreadsheet, a scanned image, an audio clip, or an HTML file (for example, `mypage.htm`).
URL	For example: `http://www.oracle.com`

Object Type	Description

	`ftp://ftp.oracle.com`
	
	`file:///D|/essbase/Docs/en/esb_infomap.htm`

Linked partition	A set of data cells that you can link to in another Essbase database.

For example, a sales manager may attach cell notes to recently updated budget items. A finance manager may link a spreadsheet containing supporting data for this quarter's results. A product manager may link bitmap images of new products. A sales manager may link the URL of a company's Web site to quickly access the information on the Web site.

Understanding LRO Types and Data Cells

LROs are linked to data cells—not to the data contained in the cells. The link is based on a member combination in the database. Adding or removing links to a cell does not affect the cell contents.

When a user links an object to a cell, Essbase stores in the object catalog information about the type of object, the name of the last user to modify the object, and the date the object was modified.

How Essbase stores the LRO depends on the LRO type:

- If the object is a cell note, the text is stored as part of the object description in the catalog entry.
- If the object is a file, Essbase stores the contents of the file in the database directory on the Essbase Server, giving it an `.lro` extension. Essbase imposes no restrictions on the data formats of linked files and performs no file-type checking. It is up to the user's client computer to render the file after retrieving it from the Essbase Server.
- If the object is a URL, Essbase stores it as part of the object description in the catalog entry. When the user tries to view the URL, Essbase does a preliminary syntax check; then the default Web browser checks for the existence of the URL.
- If the object is a linked partition, it is available through the Essbase Partitioning feature.

Before you perform tasks related to LROs, be aware of these facts:

- Essbase uses the database index to locate and retrieve linked objects. If you clear all data values from a database, the index is deleted, and so are the links to linked objects. If you restructure a database, the index is preserved, as are the links to linked objects.
- Shared members share data values but not LROs, because LROs are linked to specific member combinations, and shared members do not have identical member combinations. To link an object to shared members, link it to each shared member individually.
- You cannot change the member combination associated with any linked object. To move an object to another member combination, delete it, and then use Spreadsheet Add-in to re-link the object to the desired member combination.

Setting Up Permissions for LROs

security Users who add, edit, and delete LROs through client interfaces must have the appropriate permissions in the active database. If the object is a linked partition, the user must also have the required permissions in the database containing the linked partition. Table 29 lists the permissions required for several different tasks.

Table 29 Permissions Required for LRO Tasks

Task	Permission
Add a linked object to a database	Read-write
View an existing linked object	Read
Edit an existing linked object	Read-write
Delete a linked object	Read-write
Export the LRO catalog to a file	Read
Import the LROs from the LRO-catalog file	Read-write

To prevent users from linking files to data cells without changing user access to other data in a database, you can set the maximum file size for linked files to 1. Users can then create cell notes, link to a URL, or view linked partitions but can attach only files smaller than 1 KB.

➤ To set the maximum LRO file size for an application, see "Limiting LRO File Sizes" in the *Oracle Essbase Administration Services Online Help*.

Viewing and Deleting LROs

Users work with LROs on a cell-by-cell basis through Spreadsheet Add-in. That is, they select a cell and open the Linked Object Browser, which displays the objects linked to the selected cell. With Administration Services, you can view LROs, and you can delete all LROs for the entire database. You can also view LROs based on selection criteria such as user name and last modification date. For example, you can purge all objects older than a certain date or remove the objects belonging to a user who has left the company.

➤ To view a list of the linked objects for a database, use a tool:

Tool	Topic	Location
Administration Services	Managing LROs	*Oracle Essbase Administration Services Online Help*
MaxL	**query database**	*Oracle Essbase Technical Reference*
ESSCMD	LISTLINKEDOBJECTS	*Oracle Essbase Technical Reference*

➤ To delete the linked objects for a database, use a tool:

Tool	Topic	Location
Administration Services	Managing LROs	*Oracle Essbase Administration Services Online Help*
MaxL	**alter database**	*Oracle Essbase Technical Reference*
ESSCMD	PURGELINKEDOBJECTS	*Oracle Essbase Technical Reference*

Exporting and Importing LROs

To improve backup and data-migration capabilities, you can export and re-import LROs from data intersections in a database.

➤ To export and import linked objects for a database, use a tool:

Tool	Topic	Location
Administration Services	Exporting LROs Importing LROs	*Oracle Essbase Administration Services Online Help*
MaxL	**export lro** **import lro**	*Oracle Essbase Technical Reference*

Limiting LRO File Sizes for Storage Conservation

Because Essbase stores linked files in a repository on the server, by default, the size is unlimited. Limiting the file size prevents users from taking up too much of the server resources by storing extremely large objects. You can set the maximum linked file size for each application. If a user attempts to link a file that is larger than the limit, an error message displays.

To prevent users from attaching anything except very small files, enter 1. Setting the file size to 1 lets users link only cell notes, URLs, and files smaller than 1 KB.

Note:

The maximum file size setting applies only to linked files and does not affect cell notes or URLs. The lengths of the cell note, URL string, and LRO descriptions are fixed. For the maximum sizes of these objects, see Appendix A, “Limits.”

➤ To limit the size of a linked object, use a tool:

Tool	Topic	Location
Administration Services	Limiting LRO File Sizes	*Oracle Essbase Administration Services Online Help*

Tool	Topic	Location
MaxL	**alter application**	*Oracle Essbase Technical Reference*
ESSCMD	SETAPPSTATE	*Oracle Essbase Technical Reference*

12 Working with Typed Measures

In This Chapter

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.

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.

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 (^).
- If the measure is not designed to be aggregated, queries should be made at the same level at which data was loaded.
- Once you enable an outline to support typed measures, it cannot be reverted back to an outline that does not support typed measures.
- Text and date measures functionality applies to both aggregate storage and block storage applications.

Working with Text Measures

Text Measures Overview and Workflow

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 those five colors. Any color not represented in the finite list would be considered by Essbase to be out of range.

You create text measures at the database level. Text measures are made possible by your mapping of a set of text strings to corresponding numeric IDs. These mappings are contained in database-level text list objects that you create.

Use the following workflow to enable and use text measures:

1. In the outline properties, enable typed measures.
2. Create a text list object to store the available text values and map each text value to an ordinal number, so that Essbase can work with the text values. Optionally, map Missing and Out of Range to ordinal numbers.

   Note:

   Each numeric value can have only one text value mapped to it.
3. Create a text measure in the outline (in the Accounts dimension), and in the member properties,
   a. define it as type Text
   b. associate it with the text list object

For more information about working with text measures, see "Performing Database Operations on Text and Date Measures" on page 192.

Text List Objects and Text List Members

A cell that corresponds to a text measure can have one of a finite list of up to 1024 valid text values. Internally, Essbase needs to store the text values as numbers. Therefore, a mapping of text values to numeric values is required. You define the mapping between the text and numeric values by creating a text list object. A text list object consists of a list of text values and a numeric value that corresponds to each text value.

For example, you can create a text list object called "Customer Satisfaction Level" with the following contents:

Table 30

Name	ID
Missing	#MISSING
N/A	#OUTOFRANGE
High	1
Medium	2
Low	3

The **Name** column contains the possible text values for a text measure, and the **ID** column represents the internal numeric value used by Essbase.

Each text value must map to a unique numeric value. Any text value that does not map to an integer in the text list object is considered by Essbase to be invalid.

The first two IDs, #Missing and #OUTOFRANGE, are for handling cases where the textual data is invalid or empty. For example, if a user attempted to load an unmapped value such as "Average" to a text measure, the cell value would not be updated, and would display as #Missing in a subsequent query. If a user loads a numerical cell value which is unmapped, the subsequent query would return N/A.

Aside from #Missing and #OUTOFRANGE , all of the other IDs must be integers.

Working with Date Measures

Date Measures Overview

Date measures enable members to be associated with date-type values. The ability to process dates in the measures dimension can be useful for types of analysis that are difficult to represent using the Time dimension.

For example, an application that analyzes asset depreciation may need to track acquisition dates for a series of capital assets. The company has been in business for fifty years, so the acquisition dates span too large a period to allow for feasible Time dimension modeling (the Time dimension only covers five years).

In addition to their ability to represent values spanning large time periods, date measures can also capture date values with smaller granularity than is captured in the Time dimension, such as hours and minutes.

Implementing Date Measures

Date measures are supported for aggregate and block storage databases. You create date measures at the database level.

Use the following workflow to enable and use date measures:

1. In the outline properties, enable typed measures.
2. In the outline properties, select a date format (for example, `yyyy-mm-dd`). All date measures in the outline must use the same format.
3. Create a date measure in the outline (in the Accounts dimension), and in the member properties, define it as type Date.

For example, in ASOSamp.Sample, you can enable typed measures for the outline, select a date format, and add a measure named IntroDate defined as type Date.

The date values are stored internally as numeric values, although you load them into Essbase as formatted date strings. When queried, date measures are displayed according to the selected date format.

For more information about working with date measures, see "Performing Database Operations on Text and Date Measures" on page 192.

Functions Supporting Date Measures

The following MDX functions are useful for calculations based on date measures.

- DateDiff
- DatePart
- DateRoll
- FormatDate
- GetFirstDate
- GetLastDate
- ToDate
- ToDateEx
- Today

The following calculation functions are useful for calculations based on date measures.

- @DATEDIFF
- @DATEPART
- @DATEROLL
- @FORMATDATE
- @TODATEEX
- @TODAY

The DATEFORMAT Report Writer command

For information about these functions and commands, see the *Oracle Essbase Technical Reference*.

Performing Database Operations on Text and Date Measures

This section explains how to perform common database operations when using text and date measures.

Loading, Clearing, and Exporting Text and Date Measures

To load data to text or date measures, follow the same procedure as for loading data to members with numeric measures. The input data should contain formatted date values, or text values corresponding to the text list object that is associated with the text measure.

If you attempt to load text values that are not present in the text list object associated with that member, Essbase issues a warning message.

In aggregate storage databases, values can only be loaded at the input level; this restriction applies equally to text and date measures. In block storage databases, text and date values can be loaded at any level.

Use the following guidelines when loading text and date values into an aggregate storage database. These guidelines will help eliminate invalid aggregations.

- Use Replace mode.
- Use a single load buffer to load all values associated with date/text measures.
- Use the Aggregate_use_last aggregation method.
- Avoid loading #Missing values to text/date measures in incremental data load mode. When a #Missing value is loaded to a cell with a non-Missing value in incremental load, it is replaced with a zero value. The zero value may not have the same meaning as the #Missing value for date/text measures. Use full data load if you need to load #Missing values to date/text measures.

If mixed (numeric and text or date) data are being loaded, either ensure that Replace mode is sufficient for your numeric data, or create a separate data load process for the numeric data.

You can load text or date values with or without rules files. When a rules file is not used, you must distinguish text or date values from member names by enclosing the text values in double quotation marks and prefixing them with the string `#Txt:`.

Here is an example of a line of data in a free-form data load file:

```
"100-10" "New York" "Cust Index" #Txt:"Highly Satisfied"
```

The text value "Highly Satisfied" is pre-fixed with `#Txt:` to differentiate it from member names such as "New York".

The "#Txt" prefix is also required for date measures when a rules file is not used for data load.

You can clear, lock and send, and export text or date values just as you perform those operations on numeric values.

Aggregating Text and Date Measures

By default, text measures are assigned the “^” (no aggregation symbol) as the default operator. Text and date measures are not aggregated to higher level members along non Accounts dimensions.

If you tag a text or date measure with an operator other than “^”, it will be aggregated along other dimensions based on its internal numeric value. This is not recommended for aggregate storage databases, because only the + operator is supported, and the aggregated values likely will not have any validity for text or date measures. Additionally, Essbase does not translate out-of-range values to #OUTOFRANGE during aggregation.

For block storage databases, you can write calculation scripts that aggregate text measures in a custom fashion. You might want to aggregate text measures when they represent ranking measures. For example, consider a text list named "CustomerSatisfaction", which contains mappings such as Excellent=5, Good=4, Fair=3, Poor=2, Bad=1. The values are loaded at level-0. You can aggregate values to parent levels by taking an average of values at child levels. For example, the value of “CustomerSatisfaction” at [Qtr1] is the average of values at [Jan], [Feb], [Mar].

Retrieving Data With Text and Date Measures

Text or date measures can be queried in the same way as numerical measures, using Smart View, Grid API clients, Report Writer, and MDX. The corresponding cells are displayed with the appropriate text values or formatted date values.

The following Report Writer commands are designed to work with numeric data, and are not supported for text or date measures:

- RESTRICT
- TOP
- BOTTOM
- SORT* commands
- CALCULATE COLUMN
- CALCULATE ROW

The MDX function EnumValue and the calculation function @ENUMVALUE are designed specifically for getting the text value of text measures. These functions can be useful in MDX scripts, calc scripts, or formulas, when you need to do operations based on the text value of a member rather than its internally stored numeric value. For more information about these functions, see the *Oracle Essbase Technical Reference*

Limitations of Text and Date Measures

An outline restructure does not restructure text lists. If the mapping of numeric to text values in a text list is changed, the change will be reflected in the text data already present in the database for that text list. Therefore, when adding items to a text list, add them to the top or bottom of the list so as to avoid altering the mapping numbers of existing text list items.

Text and date measures are not supported across partitions.

Shared members and implied shared members inherit the text or date type of the original member.

Working with Format Strings

Overview of Format Strings

Using format strings, you can format the values (cell contents) of Essbase database members in numeric type measures so that they appear, for query purposes, as text, dates, or other types of predefined values. The resultant display value is the cell's formatted value (FORMATTED_VALUE property in MDX).

The underlying real value is numeric, and this value is unaffected by the associated formatted value. Format strings enable you to display more meaningful values in place of raw numeric values. For example, using a text based formatted value, you might display data cells as "High," "Medium," and "Low."

Text and date type values are additionally supported using the built-in text and date measure types. Format strings add more flexibility to your implementation, in that you can apply format strings to members in multiple dimensions, whereas with text and date measures, you can only apply one or the other to a single measures dimension. You can apply format strings to numeric dimensions; you do not have to type the dimension as text or date.

Format strings can be applied to either aggregate storage or block storage databases.

Format strings can be defined on the following members:

- All members in Measures dimension
- Members associated with explicit formula strings on other dimensions

Implementing Format Strings

Format strings are supported for aggregate and block storage databases. You implement format strings at the database level.

Use the following workflow to enable and use format strings:

1. In the outline properties, enable typed measures.
2. In the Accounts dimension, create a measure whose members you want to format, and in its member properties, edit the Associate Format String field to create an MDX format directive. For the syntax to create an MDX format directive, see "MDX Format Directive" on page 195.

MDX Format Directive

A format string is defined by the following syntax:

```
format_string_expression = MdxFormat ( string_value_expression )
```

where *string_value_expression* is a valid MDX string value expression as described in the MDX specification documented in the*Oracle Essbase Technical Reference*.

Most MDX expressions can be used to specify format strings; however, format strings cannot contain references to values of data cells other than the current cell value being formatted. The current cell value can be referenced using the MDX **CellValue** function.

Essbase treats members with invalid format strings as if there is no format string defined. Outlines can be saved with invalid format strings. Essbase generates a warning if a query consists of a member with an invalid format string.

If a member is not associated with a format string, default format rules are applied. The default format rules format a cell based on whether the measure is numeric, text, or date type. For numeric measures, the default formatted value is the text version of that number. For text measures, the default formatted value is the text value based on the associated text list object. For date values, the default format is a date string formatted according the date format string defined in the outline properties.

Functions Supporting Format Strings

The following MDX functions can be useful when applying format strings to a measure. Format strings are applied as MDX expressions, both in aggregate and block storage databases.

- **EnumText**, returns the text list label associated with the internal numeric value.
- **EnumValue**, returns the internal numeric value for a text list label.
- **CellValue**, returns the internal numeric value of the current cell.
- **NumToStr**, converts a value to a decimal string.

The **@ENUMVALUE** calculation function can be useful when writing calculation scripts for a block storage database that has text measures or format strings. This function returns the text list label associated with the internal numeric value.

The MaxL **alter session set dml_output** statement has a clause **set formatted_value on | off**. By default, formatted values are displayed in queries, but this statement can be used to turn off the display of formatted values.

The OUTFORMATTEDVALUES Report Writer command returns formatted cell values in a report.

For information about these functions, commands, and statements, see the *Oracle Essbase Technical Reference*.

Limitations of Format Strings

Format strings are not supported across partitions.

Shared members and implied shared members cannot have separate format strings: they inherit the format string of the original member.

13 Designing and Building Currency Conversion Applications

In This Chapter

The information in this chapter applies only to block storage databases and is not relevant to aggregate storage databases. Also see Chapter 57, “Comparison of Aggregate and Block Storage.”

About Currency Conversion

The Essbase currency conversion feature enables you to translate financial data from one currency into another. Currency conversion facilitates comparisons among countries and enables consolidation of financial data from locations that use different currencies.

For example, consider an organization that analyzes profitability data from the UK, reported in pounds, and from Japan, reported in yen. Comparing local currency profitability figures would be meaningless. To understand the relative contribution of each country, you must convert pounds into yen, yen into pounds, or both into another currency.

As another example, reporting total profitability for North America requires standardization of the local currency values that constitute the North America total. Assuming that the U.S., Mexico, and Canada consolidate into Total North America, the profitability total is meaningless if data is kept in local currencies. The Total North America sum is meaningful only if local currencies are converted to a common currency before consolidation.

The Essbase installation includes the option to install the Sample currency application, which consists of two databases, Interntl and Xchgrate. If you do not have access to these databases, contact your Essbase administrator. See Appendix B, “Setting Up Sample Applications.”

About the Sample Currency Application

The Sample currency application builds on the business scenario introduced in Chapter 4, “Case Study: Designing a Single-Server, Multidimensional Database,” as the Beverage Company (TBC) expands its business outside the U.S. TBC adds the following markets:

- Three locations in Canada: Toronto, Vancouver, and Montreal
- Four locations in Europe: the UK, Germany, Switzerland, and Sweden

In addition, TBC adds a new member, U.S., a consolidation of data from the U.S. regions: East, West, South, and Central.

Data for each TBC market location is captured in local currency. U.S. dollar values are derived by applying exchange rates to local values.

TBC must analyze actual data in two ways:

- Actuals are converted at actual exchange rates.
- Actuals are converted at budget exchange rates to analyze variances due to exchange rates.

After all actuals are processed, budget data is converted with budget exchange rates.

The TBC currency application consists of the main database (Interntl) and the currency database (Xchgrate). On Essbase Server, the databases are in the Sample application. If you do not have access to the databases, contact your Essbase administrator.

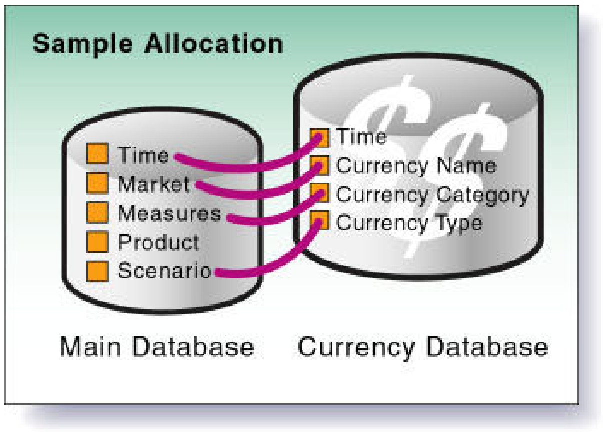
Structure of Currency Applications

In a business application requiring currency conversion, the main database is divided into at least two slices. One slice handles input of the local data, and another slice holds a copy of the input data converted to a common currency.

Essbase holds the exchange rates required for currency conversion in a separate *currency database*. The currency database outline, automatically generated by Essbase from the main database after you assign the necessary tags, typically maps a given conversion ratio onto a section of the main database. After the currency database is generated, it can be edited like any other Essbase database.

The relationship between the main database and the currency database is illustrated in Figure 44.

Figure 44 Currency Application Databases



Main Database

To enable Essbase to generate the currency database outline automatically, you modify dimensions and members in the main database outline. In the Sample currency application, the main database is Interntl.

The main database outline can contain from 3 to *n* dimensions. At minimum, the main database must contain the following dimensions:

- A dimension tagged as time. Tagging a dimension as time generates a dimension in the currency database that is identical to the time dimension in the main database. In the Sample.Interntl database, the dimension tagged as time is Year.
- A dimension tagged as accounts. Tagging a dimension as accounts and assigning currency categories to its members creates a dimension in the currency database that contains members for each of the individual currency categories. Category assignment enables the application of different exchange rates to various accounts or measures. In the Sample.Interntl database, the dimension tagged as accounts is Measures.

  Each descendant of a member inherits the currency category tag of its ancestor. A member or sub-branch of members also can have its own category.

  For example, profit and loss (P&L) accounts may use exchange rates that differ from the rates used with balance sheet accounts. In addition, some accounts may not require conversion. For example, in the Sample.Interntl database, members such as Margin% and Profit% require no conversion. You tag members not to be converted as No Conversion. The No Conversion tag is not inherited.
- A market-related dimension tagged as country. Tagging a dimension as country and assigning currency names to individual countries creates a member in the currency database for each currency. In the Sample.Interntl database, the Market dimension is tagged as country. The currency name for this dimension is USD (U.S. dollars), because all local currencies must be converted to USD, the company's common currency.

  Because multiple members can have the same currency name, the number of currency names is typically less than the total number of members in the dimension. As shown in Table 31 on page 200, the Sample.Interntl database uses only six currency names for the 15 members in the Market dimension. Each of the children of the member Europe uses a different currency and, therefore, must be assigned an individual currency name. However, the U.S. dimension and its four regional members all use the same currency. The same is true of the Canada member and its three city members. When the children of a given member share a currency, you must define a currency name for only the parent member.

Table 31 Interntl Database Currency Names

Dimensions and Members	Currency Name
Market - Country	USD (U.S. dollar)
U.S.	
East	
West	
South	
Central	
Canada	CND (Canadian dollar)
Toronto	
Vancouver	
Montreal	
Europe	
UK	GBP (British pound)
Germany	EUR (Euro)
Switzerland	CHF (Swiss franc)
Sweden	SEK (Swedish krona)

When preparing a main database outline for currency conversion, you can create an optional currency partition to tell Essbase which slice of the database holds local currency data and which holds data to be converted. The dimension that you tag as currency partition contains members for both local currency values and converted values. Local currency data is converted to common currency data using currency conversion calculation scripts. In the Sample.Interntl database, the Scenario dimension is the currency partition dimension.

For instructions on how to use currency partition dimensions, see "Keeping Local and Converted Values" on page 205.

Note:

A currency conversion partition applies only to the Currency Conversion option. It is not related to the Partitioning option that enables data to be shared between databases by using a replicated, linked, or transparent partition.

The *Oracle Essbase Spreadsheet Add-in User's Guide* provides examples of ad hoc currency reporting capabilities. Report scripts enable the creation of reports that convert data when the report is displayed, as discussed under “Converting Currencies in Report Scripts” on page 207.

Note:

For a list of methods used to create the main database outline, see “Creating Main Database Outlines” on page 203.

Currency Database

By assigning currency tags to members in the main database outline, you enable Essbase to generate the currency database automatically. In the Sample currency application, the currency database is Xchgrate.

A currency database always consists of the following three dimensions, with an optional fourth dimension:

- A dimension tagged as time, which is typically the same as the dimension tagged as time in the main database. This allows the currency database to track currency fluctuations over time and to accurately convert various time slices of the main database. In the Sample.Xchgrate database, the dimension tagged as time is Year.

  Each member of the time dimension in the main database must be defined in the currency database. Values by time period in the main database usually are converted to the exchange rates of their respective time period from the currency database (although you can convert data values against the exchange rate of any period).

- A dimension tagged as country, which contains the names of currencies relevant to the markets (or countries) defined in the main database. Each currency name defined in the main database must also exist in the currency database. The currency names define the country-to-exchange rate mapping when conversion occurs.

  In the Sample.Xchgrate database, the country dimension is CurName. Table 32 lists the currency names in the CurName dimension:

Table 32 Xchgrate Database Currency Names

Dimension and Members	Alias Name
CurName - Country	U.S. dollar
USD	Canadian dollar
CND	British pound
GBP	Euro
EUR	Swiss franc
CHF	Swedish krona
SEK	

- A dimension tagged as accounts, which enables the application of various rates to members of the dimension tagged as accounts in the main database. The categories defined for the accounts dimension in the main database are used to form the members in the accounts dimension of the currency database. For example, it may be necessary to convert Gross Profit and Net Profit using one category of rates, while other accounts use a different set of rates.

In the Sample.Xchgrate database, the dimension tagged as accounts is CurCategory, and the account categories included are P&L (Profit & Loss) and B/S (Balance Sheet).

- A currency database, which typically includes an optional currency type dimension, which enables different scenarios for currency conversion. Typically, an application has different exchange rates for different scenarios, such as actual, budget, and forecast. To convert data between scenarios, select which type of rate to use.

  The currency type dimension is created when you generate the currency outline and is not directly mapped to the main database. Therefore, member names in this dimension need not match member names of the main database.

  In the Sample.Xchgrate database, the currency type dimension is CurType. CurType includes actual and budget scenarios.

Note:

For information about creating the currency database outline, see "Building Currency Conversion Applications and Performing Conversions" on page 203.

Conversion Methods

Different currency applications have different conversion requirements. Essbase supports two conversion methods:

- Overwriting local values with converted values.

  Some applications require that only converted values be stored in the main database. Local values are entered, and the conversion operation overwrites local values with common currency values. This method assumes that there is no requirement for reporting or analyzing local currencies.

  Because this operation overwrites data, load local values and recalculate the data each time you perform a conversion. This method is useful only for single (not ongoing) conversions.

- Keeping local and converted values.

  Most applications require that data be stored in both local and common currency (converted) values. This method permits reporting and analyzing local data, and data modifications and recalculations are easier to control. To use this method, define a currency partition (see "Main Database" on page 199).

Either of these methods may require a currency conversion at report time. Report time conversion enables analysis of various exchange rate scenarios without actually storing data in the database. The currency conversion module enables performance of ad hoc conversions. You perform ad hoc conversions by using Spreadsheet Add-in, as discussed in the *Oracle Essbase Spreadsheet Add-in User's Guide*, or by using a report script, as discussed under "Converting Currencies in Report Scripts" on page 207.

Building Currency Conversion Applications and Performing Conversions

To build a currency conversion application and perform conversions, use the following process:

1. Create or open the main database outline.

   See "Creating Main Database Outlines" on page 203.

2. Prepare the main database outline for currency conversion.

   See "Preparing Main Database Outlines" on page 203.

3. Generate the currency database outline.

   See "Generating Currency Database Outlines" on page 204.

4. Link the main and currency databases.

   See "Linking Main and Currency Databases" on page 204.

5. Convert currency values.

   See "Converting Currency Values" on page 204.

6. Track currency conversions.

   See "Tracking Currency Conversions" on page 208.

7. If necessary, troubleshoot currency conversion.

   See "Troubleshooting Currency Conversion" on page 210.

Creating Main Database Outlines

To create a main database outline, create or open an Essbase database outline, modify the outline as needed, and save the outline for use in the currency conversion application.

➤ To create an outline or open an existing outline, use a tool:

Tool	Topic	Location
Administration Services	Opening and Editing Outlines	*Oracle Essbase Administration Services Online Help*
MaxL	**create database**	*Oracle Essbase Technical Reference*
ESSCMD	CREATEDB	*Oracle Essbase Technical Reference*

Preparing Main Database Outlines

After you create or open the main database outline, modify dimensions and members to enable Essbase to generate the currency database outline automatically. See "Main Database" on page 199.

➤ To prepare a main database outline, see “Preparing the Main Database Outline for Currency Conversion” in the *Oracle Essbase Administration Services Online Help*.

Generating Currency Database Outlines

After you verify and save the main database outline, you can generate the currency outline. The currency outline contains dimensions, members, currency names, and currency categories previously defined in the main database outline. The currency database outline is basically structured and ready to use after being generated but may require additions to make it complete.

➤ To generate a currency database outline, see “Generating a Currency Database Outline” in the *Oracle Essbase Administration Services Online Help*.

Linking Main and Currency Databases

To perform a currency conversion calculation, Essbase must recognize a link between the main and currency databases. Generating a currency outline does not automatically link a main database with a currency database. When you link the databases, you specify the conversion calculation method and the default currency type member.

➤ To link main and currency databases, see “Linking a Database to a Currency Database” in the *Oracle Essbase Administration Services Online Help*.

Converting Currency Values

After you create a currency conversion application, you convert data values from a local currency to a common, converted currency by using the CCONV command in calculation scripts. For example, you might convert data from a variety of currencies into USD (U.S. dollars). You can convert the data values back to the original, local currencies by using the CCONV TOLOCALRATE command.

You can convert all or part of the main database using the rates defined in the currency database. You can overwrite local values with converted values, or you can keep both local and converted values in the main database, depending on your tracking and reporting needs.

Note:

When running a currency conversion, ensure that the data being converted is not simultaneously being updated by other user activities (for example, a calculation, data load, or currency conversion against the same currency partition). Concurrent activity on the data being converted may produce incorrect results. Essbase does not display a warning message in this situation.

Note:

When you convert currencies using the CCONV command, the resulting data blocks are marked as dirty for the purposes of Intelligent Calculation. Thus, Essbase recalculates all converted blocks when you recalculate the database.

To see sample currency conversion calculation scripts, see the *Oracle Essbase Technical Reference*.

Overwriting Local Values with Converted Values

To overwrite local values, you need *not* create a currency partition dimension in the main database. Use the CCONV command in a calculation script to convert all data in the database:

The following calculation script converts the values in the database to USD:

```
CCONV USD;
CALC ALL;
```

If required, you can specify a currency name that contains the required exchange rate. The following calculation script converts the values in the database to USD, using the exchange rate for Jan as defined in the currency database:

```
CCONV Jan->USD;
CALC ALL;
```

The CALC ALL command is required in the examples shown because the CCONV command converts only currencies. It does not consolidate or calculate members in the database.

The following calculation script uses the "Act xchg" rate to convert the converted values back to their original local currency values:

```
CCONV TOLOCALRATE "Act xchg";
CALC ALL;
```

Note:

You cannot use the FIX command unless you are using a currency partition dimension and the CCTRACK setting is TRUE in the `essbase.cfg` file.

Keeping Local and Converted Values

You can keep local and converted values in a database. In the main database, you must define the members that store the local and converted values by creating a currency partition dimension (see "Main Database" on page 199). The currency partition dimension has one partition for local values and one for converted values.

➤ To create a calculation script that copies local data to a converted partition and calculates the data:

1 **Use the DATACOPY command to copy data from the local to the converted partition.**

2 **Use the FIX command to calculate only the converted partition and use the CCONV command to convert the data.**

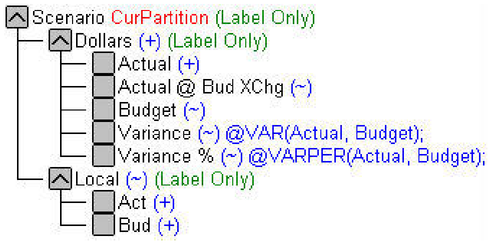
Note:

When using a currency partition dimension, you must FIX on a member of the dimension to use the CCONV command.

3 **Use the CALC command to recalculate the database.**

The following example is based on the Sample.Interntl database and the corresponding Sample.Xchgrate currency database. Figure 45 shows the currency partition from the Sample.Interntl database.

Figure 45 Calculating Local and Converted Currency Conversions



```
Scenario CurPartition (Label Only)
  Dollars (+) (Label Only)
    Actual (+)
    Actual @ Bud XChg (~)
    Budget (~)
    Variance (~) @VAR(Actual, Budget);
    Variance % (~) @VARPER(Actual, Budget);
  Local (~) (Label Only)
    Act (+)
    Bud (+)
```

The following calculation script performs three currency conversions for Actual, Budget, and Actual @ Bud Xchg data values:

```
/* Copy data from the local partition to the master partition (for
converted values) */
DATACOPY Act TO Actual;
DATACOPY Bud TO Budget;
/* Convert the Actual data values using the "Act xchg" rate */
FIX(Actual)
   CCONV "Act xchg"->US$;
ENDFIX

* Convert the Budget data values using the "Bud xchg" rate */
FIX(Budget)
   CCONV "Bud xchg"->US$;
ENDFIX

/* Convert the "Actual @ Bud XChg" data values using the
"Bud xchg" rate */
FIX("Actual @ Bud XChg")
   CCONV "Bud xchg"->US$;
ENDFIX

/* Recalculate the database */
CALC ALL;
CALC TWOPASS;
```

The following calculation script converts the Actual and Budget values back to their original local currency values:

```
FIX(Actual)
   CCONV TOLOCALRATE "Act xchg";
ENDFIX
FIX(Budget)
   CCONV TOLOCALRATE "Bud xchg";
ENDFIX
CALC ALL;
```

Note:

When you convert currencies using the CCONV command, the resulting data blocks are marked as dirty for the purposes of Intelligent Calculation. Thus, Essbase recalculates all converted blocks when you recalculate the database.

Calculating Databases

If you execute a CALC ALL command to consolidate the database after running a conversion, meaningful total-level data is generated in the converted base rate partition, but the local rate partition contains a meaningless consolidation of local currency values. To prevent meaningless consolidation, use the calculation command SET UPTOLOCAL, which restricts consolidations to parents with the same defined currency. For example, all cities in the U.S. use dollars as the unit of currency. Therefore, all children of U.S. consolidate to U.S. Consolidation stops at the country level, however, because North America contains countries that use other currencies.

Converting Currencies in Report Scripts

You can convert currencies in report scripts, using the CURRENCY command to set the output currency and the currency type. For the syntax and definitions of Report Writer commands, see the *Oracle Essbase Technical Reference*.

Note:

Essbase cannot perform ”on the fly” currency conversions across transparent databases. If two transparent partition databases are calculated using different conversions, you cannot perform currency conversions in reports.

The following Sample report contains first-quarter Budget Sales for colas, using the January exchange rate for the Peseta currency.

```
                       Illinois Sales Budget

                     Jan        Feb        Mar
                     ========   ========   ========
    100-10              3          3          3
    100-20              2          2          2
    100-30           #Missing   #Missing   #Missing
    100                 5          5          5
             Currency: Jan->Peseta->Act xchg

             Currency: Jan->Peseta->Act xchg
```

```
                              Illinois Sales Budget

                            Jan         Feb         Mar
                            ========    ========    ========
     100-10                   3           3           3
     100-20                   2           2           2
     100-30                 #Missing    #Missing    #Missing
     100                      5           5           5
```

Use the following script to create the Sample currency conversion report:

```
<Page (Market, Measures, Scenario)
{SupCurHeading}
Illinois Sales Budget
       <Column (Year)
       <children Qtr1
<Currency "Jan->Peseta->Act xchg"
<Ichildren Colas
   !
{CurHeading}
Illinois Sales Budget
       <Column (Year)
       <children Qtr1
   !
```

Tracking Currency Conversions

You can use the CCTRACK setting in the `essbase.cfg` file to control whether Essbase tracks the currency partitions that have been converted and the exchange rates that have been used for the conversions. Tracking currency conversions has the following advantages:

- Enables conversion to occur at report time through Spreadsheet Add-in or Report Writer
- Enables conversion of a converted currency back to its original, local rate using the CONV TOLOCALRATE command
- Prevents data inaccuracies due to accidental reconversion of data during a currency calculation

By default, CCTRACK is turned on. Essbase tracks which currency partitions have been converted and which have not. The tracking is done at the currency partition level: a database with two partitions has two flags, each of which can be "converted" or "unconverted." Essbase does not store a flag for member combinations within a partition.

When using a currency partition, and when CCTRACK is set to TRUE (the default) in `essbase.cfg`, you must FIX on a single currency partition member. You cannot FIX on multiple members, because CCTRACK works at the currency partition member level and marks as converted or unconverted all data associated with the currency partition member. For example, in the Sample.Basic database, the following example is valid:

```
FIX(Actual)
CCONV "Act xchg"->US$;
ENDFIX
```

In the Sample.Basic database, if you were able to use a FIX command to convert the actual values for only the members Jan and Feb, the database would have converted and unconverted data in the same currency partition, causing a data consistency issue.

Reasons to Turn Off CCTRACK

For increased efficiency when converting currency data between currency partitions, consider turning off CCTRACK. For example, if you load data for the current month into the local partition, use the DATACOPY command to copy the entire currency partition that contains the updated data, and then run the conversion on the currency partition.

Note:

Always do a partial data load to the local partition and use the DATACOPY command to copy the *entire* currency partition to the converted partition before running the currency conversion. Updating data directly into the converted partition causes incorrect results.

Methods for Turning Off CCTRACK

You can turn off CCTRACK in three ways:

- Use the SET CCTRACKCALC ON|OFF command in a calculation script to turn off CCTRACK temporarily. You can use this command at calculation time to increase flexibility and efficiency during currency conversion.
- Use the CLEARCCTRACK calculation command to clear the internal exchange rate tables created by CCTRACK. You can use the command inside a FIX statement to clear the exchange rates for a currency partition. Use the command after a data load to reset the exchange rate tables so that they are ready for future currency conversion calculations.
- Set CCTRACK to FALSE in the `essbase.cfg` file. Setting CCTRACK to False turns off the tracking system and has the following results:
    - The CCONV command assumes that data is unconverted (is in local currency). If you accidentally run the CCONV command multiple times on the same data, the resulting data is inaccurate.
    - Similarly, the currency report options assume that the data is unconverted (is in local currency). If the data already has been converted in the database, it is reconverted at report time, resulting in inaccurate data.
    - The restrictions on using the FIX and DATACOPY commands in currency conversions do not apply.

    Note:

    When running a currency conversion, ensure that the data being converted is not simultaneously being updated by other user activities (for example, a calculation, data load, or currency conversion against the same currency partition). Concurrent activity on the data being converted may produce incorrect results. Essbase does not display a warning message in this situation.

Troubleshooting Currency Conversion

See “Troubleshooting Currency Conversion” in the *Oracle Essbase Administration Services Online Help*.

14 Designing Partitioned Applications

In This Chapter

Understanding Essbase Partitioning

A partition is the region of a database that is shared with another database. An Essbase partitioned application can span multiple servers, processors, or computers.

Partition Types

Table 33 lists the types of partitions that are supported in Essbase:

Table 33 Partition Types

Partition Type	Description	Applies To
Replicated	A copy of a portion of the data source that is stored in the data target. See "Replicated Partitions" on page 220.	Block storage databases Aggregate storage databases
Transparent	Allows users to access data from the data source as though it were stored in the data target. The data is, however, stored at the data source, which can be in another application or Essbase database, or on another Essbase Server. See "Transparent Partitions" on page 224.	Block storage databases Aggregate storage databases
Linked	Sends users from a cell in one database to a cell in another database. Linked partitions give users a different perspective on the data. See "Linked Partitions" on page 230.	Block storage databases

Use the information in Table 34 to help you choose which type of partition to use:

Table 34 Features Supported by Partition Type

Feature	Replicated	Transparent	Linked
Up-to-the-minute data		x	x
Reduced network traffic	x		x
Reduced disk space		x	x
Increased calculation speed	x		
Smaller databases		x	x
Improved query speed	x		x
Invisible to end users	x	x	
Access to databases with different dimensionality			x
Easier to recover	x		
Less synchronization required			x
Ability to query data based on its attributes		x	x
Ability to use front-end tools that are not distributed OLAP-aware	x	x	
Easy to perform frequent updates and calculations		x	
Ability to update data at the data target		x	x
View data in a different context			x
Perform batch updates and simple aggregations	x		

Parts of a Partition

Partitions contain the following parts, as illustrated in Figure 46 and described in Table 35 on page 213.

Figure 46 Parts of a Partition

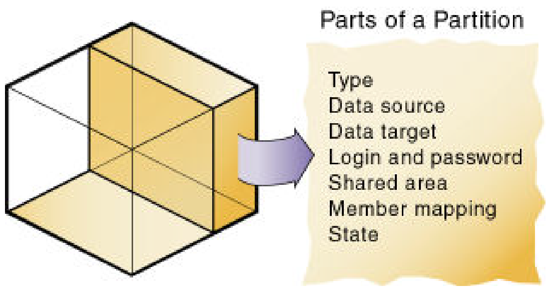


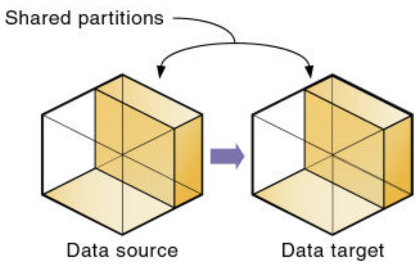
Table 35 Parts of a Partition

Part	Description
Type of partition	A flag indicating whether the partition is replicated, transparent, or linked.
Data source information	The server, application, and database name of the data source.
Data target information	The server, application, and database name of the data target.
Login and password	The login and password information for the data source and the data target. This information is used for internal requests between the two databases to execute administrative and end-user operations.
Shared areas	A definition of one or more areas, or regions, shared between the data source and the data target. To share more than one noncontiguous portion of a database, define multiple areas in a single partition. This information determines which parts of the data source and data target are shared so that Essbase can put the proper data into the data target and keep the outlines for the shared areas synchronized.
Member mapping information	A description of how the members in the data source map to members in the data target. Essbase uses this information to determine how to put data into the data target if the data target and the data source use different names for some members and dimensions.
State of the partition	Information about whether the partition is up-to-date and when the partition was last updated.

Data Sources and Data Targets

Partitioned databases contain at least one data source (the primary site of the data) and at least one data target (the secondary site of the data). One database can serve as the data source for one partition and the data target for another partition. When defining a partition, you map cells in the data source to their counterparts in the data target.

Figure 47 Data Source and Data Target



An Essbase database can contain many partitions, as well as data that is not shared with any other Essbase database. You can define partitions between the following databases:

- Different databases in different applications, as long as each database uses the same language and the same Unicode-related mode.

  The applications can be on the same computer or different computers.

- Different databases in one block storage application.

  This practice is not recommended, because the full benefits of partitioning databases are realized when each database is in a separate application.

You can define only one partition of each type between the same two databases. For example, you can create only one replicated partition between the Sampeast.East and Samppart.Company databases. The East or Company databases can, however, contain many replicated partitions that connect to other databases.

One database can serve as the data source or data target for multiple partitions. To share data among many databases, create multiple partitions, each with the same data source and a different data target, as shown in Figure 48:

Figure 48 Data Shared at Multiple Targets

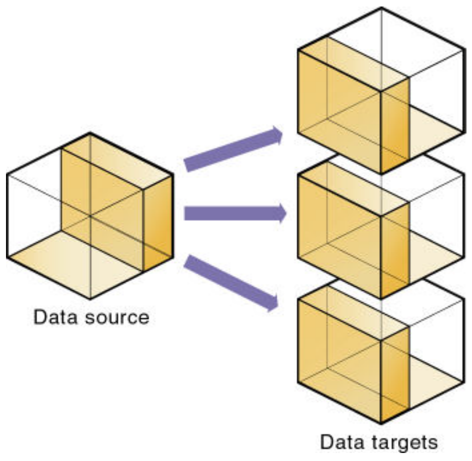


Table 36 lists the combinations of block storage and aggregate storage databases as data target and data source that are supported by each partition type:

Table 36 Combinations of Data Targets and Data Sources Supported by Partition Type

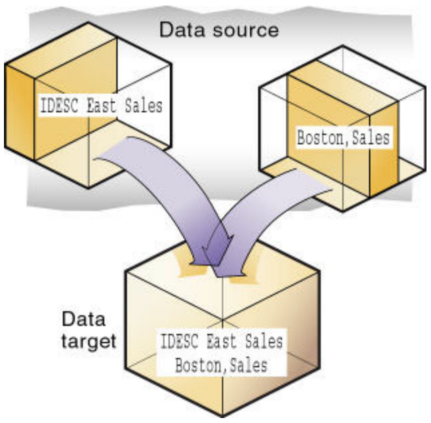
Target	Source	Replicated	Transparent	Linked
Block storage	Block storage	Yes	Yes	Yes
Block storage	Aggregate storage	No	Yes	Yes
Aggregate storage	Aggregate storage	No	Yes	Yes
Aggregate storage	Block storage	Yes	Yes	Yes

Overlapping Partitions

An overlapping partition occurs when similar data from multiple databases is the data source for one data target in a partition.

For example, IDESC East, Sales from database 1 and Boston, Sales from database 2 are mapped to IDESC East, Sales and Boston, Sales in database 3. Because Boston is a member of the East dimension, the data for Boston mapped to database 3 from database 1 and database 2 overlap. This data overlap results in an overlapping partition, as shown in Figure 49:

Figure 49 Overlapping Partitions



An overlapping partition is allowed in linked partitions but is invalid in replicated and transparent partitions and generates an error message during validation.

Substitution Variables in Partition Definitions

Using substitution variables in partition definitions enables you to base the partition definition on different members at different times. Substitution variables act as global placeholders for information that changes regularly; each variable has an assigned value, which the Database Manager can change anytime. For example, you can define a substitution variable named Curmonth and change the substitution variable value to the member name for each month throughout the year to Jan, Feb, Mar, and so on. In this example, using a substitution variable reduces the partition size because you need not include all months in the partition definition area to access data from one month.

To specify a substitution variable in an area definition or in a mapping specification, use the "Use text editor" or "Use inline editing" editing option. Insert "&" at the beginning of the substitution variable name; for example, &Month. Essbase uses substitution values when you verify the partition. When you perform any process that uses partitioned data, Essbase resolves the substitution variable name to its value. The substitution variable name is displayed when you view the partition definition. See "Using Substitution Variables" on page 120.

Attributes in Partitions

For block storage databases, you can use attribute functions for partitioning on attribute values, but you cannot partition an attribute dimension. Use attribute values to partition a database to access members of a dimension according to their characteristics.

For example, in the Sample.Basic database, you cannot partition the Pkg Type attribute dimension, but you can create a partition that contains all the members of the Product dimension that are associated with either or both members (Bottle and Can) of the Pkg Type dimension.

If you create a partition that contains members associated with Can, you can access data only on Product members that are packaged in cans; namely, 100-10, 100-20, and 300-30.

You can use the @ATTRIBUTE command and the @WITHATTR command to define partitions.

For example, to extract data on all members of the Product dimension that are associated with the Caffeinated attribute dimension, you can create a partition such as @ATTRIBUTE (Caffeinated). But you cannot partition the Caffeinated attribute dimension.

Based on the previous example, this partition is correct:

```
Source                    Target
@ATTRIBUTE(Caffeinated)   @ATTRIBUTE(Caffeinated)
```

This partition is incorrect:

```
Source                    Target
Caffeinated               Caffeinated
```

For more information about these commands, see the *Oracle Essbase Technical Reference*.

Also see Chapter 10, “Working with Attributes.”

Version and Encoding Requirements

Version: Both ends (the source and target) of the partition must be on the same release level of Essbase Server for these partition types:

- Replicated
- Transparent
- Linked

Encoding: The application mode—Unicode mode or non-Unicode mode—of both ends of the partition must be the same for these partition types:

- Replicated
- Transparent

Partition Design Requirements

Use the information in this section to carefully design partitions before implementing them.

Benefits of Partitioning

Partitioning can provide the following benefits:

- For block storage databases, data synchronization across multiple databases

  Essbase tracks changes made to data values in a partition and provides tools for updating the data values in related partitions.

- Outline synchronization across multiple databases (except when an aggregate storage database is the target of a transparent partition)

  Essbase tracks changes made to the outlines of partitioned databases and provides tools for updating related outlines.
- Ability for user navigation between databases with differing dimensionality

  When users drill across to the new database, they can drill down to more-detailed data.

Partitioning Strategies

Based on user requirements, select a partitioning strategy:

- Partition applications from the top down.

  Use *top-down partitioning* to split a database onto multiple processors, servers, or computers, which can improve the scalability, reliability, and performance of databases. To achieve the best results with top-down partitioning, create a separate application for each partitioned database.
- Partition applications from the bottom up.

  Use *bottom-up partitioning* to manage data flow between multiple related databases, which can improve the quality and accessibility of the data in databases.
- Partition databases according to attribute values associated with base dimensions (a standard dimension associated with one or more attribute dimensions).

  Use this strategy to extract data based on the characteristics of a dimension, such as flavor or size.

  Note:

  You cannot partition attribute dimensions. See "Attributes in Partitions" on page 215.

Guidelines for Partitioning a Database

Use the following information to help you determine whether to partition a database.

- Partition a database when:
  - The data should be closer to the people who are using it.
  - A single failure would be catastrophic.
  - It takes too long to perform calculations after new data is loaded, and you want to improve performance by spreading calculations across multiple processors or computers.
  - Users want to see the data in different application contexts, and you want to control how users navigate between databases.
  - You need to synchronize information from different sources.

    - You plan to add new organizational units that would benefit from having their own databases.
    - Users have to wait as other users access the database.
    - You want to save disk space by giving users access to data stored in a remote location.
    - You want to reduce network traffic by replicating data in several locations.
    - You need to control database outlines from a central location.
    - You need client write-back functionality on an aggregate storage database.

      See "Using a Transparent Partition to Enable Write-Back for Aggregate Storage Databases" on page 936.
- Do not partition a database when:
    - You have disk space, network bandwidth, and administrative resource concerns.
    - You perform complex allocations where unit level values are derived from total values.
    - You are required to keep all databases online at all times.

      Keeping databases online can be a problem if you have databases in several time zones, because peak user load may differ between time zones. Using linked and transparent partitions exacerbate this problem, but using replicated partitions might help.
    - Databases are in different languages or Unicode-related modes.

      Essbase can partition databases only if each database uses the same language, or each database uses the same Unicode or non-Unicode mode.

Guidelines for Partitioning Data

When designing a partitioned database, use the following information to help you determine which data to include in each partition:

- Which database should be the data source and which the data target? The database that "owns" the data, where the data is updated and where most of the detail data is stored, should be the data source.
- Are some parts of the database accessed more frequently than others?
- What data can you share among sites?
- How granular must the data be at each location?
- How frequently is the data accessed, updated, or calculated?
- What are the available resources: disk space, CPUs, and network resources?
- How much data must be transferred over the network? How long does it take?
- Is the data stored in one or multiple locations?
- Is the data accessed in one or multiple locations?
- Is there information in separate databases that should be accessed from a central location? How closely are groups of data related?

See "Case Studies for Designing Partitioned Databases" on page 232.

Security for Partitioned Databases

Users accessing replicated, transparent, or linked partitions may need to view data stored in multiple databases. The following sections describe how to set up security so that users do not view or change inappropriate data.

Setting up End-User Security

Create the required end users with the correct filters.

1. Create accounts for users at the data target.

   See "Managing Users and Groups in Native Security Mode" on page 630.

2. Create read and write filters at the data target to determine what end users can view and update.

   See Chapter 38, "User Management and Security."

3. If you are creating a replicated partition, determine whether users can make changes to a replicated partition at the data target. This setting overrides user filters that allow users to update data.

   See the *Oracle Essbase Administration Services Online Help*.

4. If you are creating a linked partition, create accounts for users at the data source. Users accessing linked databases may need to connect to multiple databases.

   See "Drill Across and Linked Partitions" on page 231.

Setting up Administrator Security

The administrative account performs all read and write operations requested by the data target for the data source. For example, when end users request data at the data target, the administrative account retrieves the data. When end users update data at the data target, the administrative account logs into the data source and updates the data there.

You can create filters on the administrative account in addition to filters on the end users. Filters on the administrative account can ensure that no one at the data target can view or update inappropriate data. For example, the administrator at the corporate database can restrict write access on certain cells to avoid relying on administrators in the regions to set up security correctly for each end user.

Create the required administrative users with the correct filters.

1. Create an administrative account at the data source and data target.

   See "Setting the User Name and Password" on page 239.

   Essbase uses this account to log onto the data source to retrieve data and to perform outline synchronization operations.

2. Create read and write filters to determine which data administrators can view and update.

   See Chapter 38, “User Management and Security”.

   - For replicated partitions, set up read filters at the data source to determine which data Essbase reads when replicating, and set up write filters at the data target to determine which data Essbase writes to when replicating.
   - For transparent partitions, set up read filters at the data source to determine which data Essbase retrieves for end users, and set up write filters at the data source to determine which data Essbase updates for end users.

Using Backup and Restore and Transaction Logging and Replay with Partitioned Databases

If you are using Essbase backup and restore and transaction logging and replay features with partitioned databases, there are guidelines that you must follow. See the *Oracle Hyperion Enterprise Performance Management System Backup and Recovery Guide*.

Replicated Partitions

A replicated partition is a copy of a portion of the data source that is stored in the data target. Some users can then access the data in the data source while others access it in the data target.

For example, in the Samppart and Sampeast sample applications, the DBA at The Beverage Company (TBC) created a replicated partition between the East database and the Company database containing Actual, Budget, Variance, and Variance%. Users in the eastern region now store their budget data locally. Because they do not have to retrieve this data live from corporate headquarters, response times are faster, and they have more control over the downtimes and administration of local data. See “Case Study 1: Partitioning an Existing Database” on page 232.

Changes to the data in a replicated partition flow from the data source to the data target. Changes made to replicated data in the data target do not flow back to the data source. If users change the data at the data target, Essbase overwrites their changes when the DBA updates the replicated partition.

When a replicated partition is defined, the DBA can select a setting to prevent the data in the replicated portion of the data target from being updated. This setting takes precedence over access provided by security filters and is also honored by batch operations, such as data load and calculation. By default, a target of a replicated partition cannot be updated. To set a partition so that it can be updated, see the *Oracle Essbase Administration Services Online Help*.

Use a replicated partition to achieve any of the following goals:

- Decrease network activity
- Decrease query response times
- Decrease calculation times
- Recover more easily from system failures

Rules for Replicated Partitions

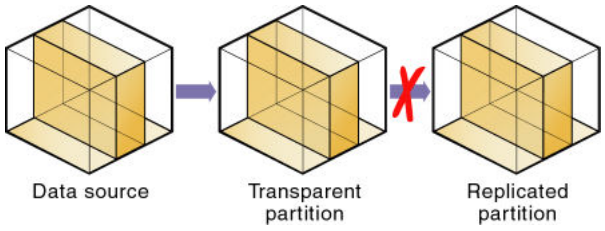
Replicated partitions must follow these rules:

- You must be able to map the shared replicated areas of the data source and data target outlines, although the shared areas need not be identical. You must tell Essbase how each dimension and member in the data source maps to each dimension and member in the data target.

  The data source and data target outlines for the non-shared areas do not have to be mappable.

- Because none of the areas that you use as a replicated partition target can come from a transparent partition source, you cannot create a replicated partition on top of a transparent partition, as shown in Figure 50:

  Figure 50 Invalid Replicated Partition

- The cells in the data target of a replicated partition cannot come from two data sources; the cells in one partition must come from one database. To replicate cells from more than one database, create a different partition for each data source.

  The cells in a data target can be the data source for a different replicated partition. For example, if the Samppart.Company database contains a replicated partition from the Sampeast.East database, you can replicate the cells in Sampeast.East into a third database, such as Sampwest.West.

- You cannot use attribute members to define a replicated partition. For example, associated with the Market dimension, the Market Type attribute dimension members are Urban, Suburban, and Rural. You cannot define a partition on Urban, Suburban, or Rural, because a replicated partition contains dynamic data, not stored data. Therefore, an attempt to map attributes in replicated partitions results in an error message. However, you can use the WITHATTR command to replicate attribute data.

Advantages of Replicated Partitions

- Because data is stored closer to end users, in the data target, replicated partitions can decrease network activity, resulting in improved retrieval times for users.
- The data is more easily accessible to all users. Some users access the data at the data source, others at the data target.
- Failures are not as catastrophic. Because the data is in more than one place, if one database fails, only the users connected to that database are unable to access the information. Data is still available at and can be retrieved from the other sites.

- Local DBAs can control the downtime of their local databases. For example, because users in the eastern region are accessing their own replicated data instead of the Company database, DBAs can bring down the Company database without affecting users in the eastern region.
- Because only the relevant data is kept at each site, databases can be smaller. For example, users in the eastern region can replicate only the eastern budget information, instead of accessing a larger company database containing budget information for all regions.

Disadvantages of Replicated Partitions

- You need more disk space because data is stored in multiple locations.
- Because the DBA must manually refresh data regularly, users may not see the latest version of the data.

Performance Considerations for Replicated Partitions

To improve the performance of replicated partitions, follow these guidelines:

- Do not replicate members that are dynamically calculated in the data source, because Essbase must probe the outline to find dynamically calculated members and their children to determine how to perform the calculation.
- Do not replicate derived data from the data source. Instead, replicate the lowest practical level of each dimension and perform the calculations on the data target after you complete the replication.

  For example, to replicate the database along the Market dimension:

  - Define the shared area as the lowest-level members of the Market dimension that you care about, for example, East, West, South, and Central and the level 0 members of the other dimensions.
  - After you complete the replication, calculate the values for Market and the upper-level values in the other dimensions at the data target.

    Sometimes you cannot calculate derived data at the data target. In that case, replicate it from the data source. For example, you cannot calculate derived data at the data source if the data meets any of the following criteria:

    - Requires that data outside the replicated area be calculated.
    - Requires calculation scripts from which you cannot extract only the portion to be calculated at the data target.
    - Is being replicated onto a computer with little processing power, such as a laptop.
- To optimize the replication of an aggregate storage database when the aggregate storage database is the target and a block storage database is the source and the two outlines are identical, use one of these methods:

- The REPLICATIONASSUMEIDENTICALOUTLINE configuration setting in `essbase.cfg`. The setting can be enabled at the server, application, or database level. The syntax for the setting is as follows:

  ```
  REPLICATIONASSUMEIDENTICALOUTLINE [appname [dbname]] TRUE | FALSE
  ```

  When updating the `essbase.cfg` file, you must stop and then restart Essbase Server for the changes to take effect.

- The **alter database** MaxL statement with the **replication_assume_identical_outline** grammar. The statement can be enabled only at the database level. The syntax for the statement is as follows:

  ```
  alter database appname.dbname enable | disable
  replication_assume_identical_outline;
  ```

  When using the **alter database** statement, you do not need to stop and restart the aggregate storage application.

Both optimization methods affect only the target aggregate storage application; the source block storage application is not affected. The methods do not apply to block storage replication.

- Partitioning along a dense dimension takes longer than partitioning along a sparse dimension. When Essbase replicates data partitioned along a dense dimension, it must access every block in the data source and then create each block in the data target during the replication operation.
- You cannot replicate data into a member that is dynamically calculated at the data target. Essbase does not load or replicate into Dynamic Calc and Dynamic Calc and Store members, because these members do not contain data until a user requests it at runtime. Essbase avoids sending replicated data for both dynamic dense and dynamic sparse members on the replication target, because this data is not stored on the data target.

  To replicate only the data values that have changed instead of the entire partition, see "Populating or Updating Replicated Partitions" on page 251.

Replicated Partitions and Port Usage

With replicated partitions, users connect to the target database only. When data is updated on the target database, the process of replicating data from the source database to the target database utilizes one port and this connection is based on the user name declared in the partition definition (partition user).

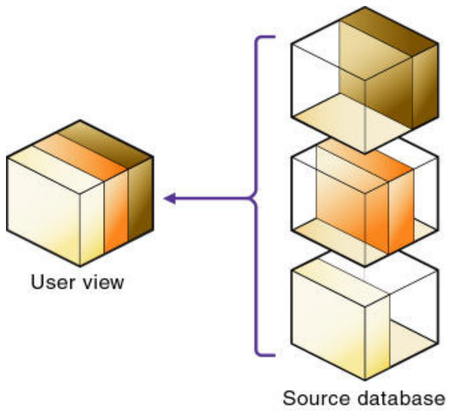
Note:

Because of the short-term nature of replication, replicated partitions and ports are rarely a problem.

Transparent Partitions

A transparent partition allows users to manipulate data that is stored remotely as if it were part of the local database. The remote data is retrieved from the data source each time that users at the data target request it. Users do not need to know where the data is stored, because they see it as part of their local database.

Figure 51 Transparent Partitions



Because data is retrieved directly from the data source, users see the latest version. When they update the data, their updates are written back to the data source. This process means that other users at the data source and the data target have immediate access to those updates.

With a transparent partition, users at the data source and at the data target may notice slower performance as more users access the source data.

For example, the DBA at TBC can use a transparent partition to calculate each member of the Scenario dimension on a separate computer. This process reduces the elapsed time for the calculation while providing users with the same view of the data. See "Case Study 1: Partitioning an Existing Database" on page 232.

Use a transparent partition to achieve the following goals:

- Show users the latest version of the data
- Allow users at the data target to update data
- Decrease disk space

Rules for Transparent Partitions

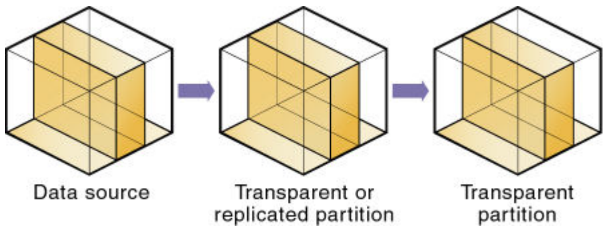
Transparent partitions must follow these rules:

- The shared transparent areas of the data source and data target outlines need not be identical, but you must be able to map the dimensions in them. You must tell Essbase how each dimension and member in the data source maps to each dimension and member in the data target.
- The data source and data target outlines for the nonshared areas need not be mappable, but attribute associations must be identical. Otherwise, users can get incorrect results for some

retrievals. For example, if product 100-10-1010 is associated with the Grape Flavor attribute on the source, but product 100-10-1010 is not associated with Grape on the target, the total of sales for all Grape flavors in New York is incorrect.

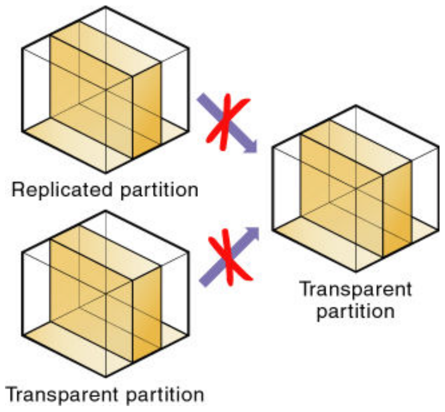
- The partition definition must contain only stored members. You cannot use attribute dimensions or members to define a transparent partition. For example, the Market Type attribute dimension, which is associated with the Market dimension, has members Urban, Suburban, and Rural. You cannot define a partition on Urban, Suburban, or Rural.
- If a cell is mapped from the data source to an aggregate storage database as the target, all the cell's dependents must also be mapped to the same partition definition.
- You can create a transparent partition on top of a replicated partition. In other words, you can create a transparent partition target using a replicated partition source, as shown in Figure 52

Figure 52 Valid Transparent Partition



- As illustrated in Figure 53, you cannot create a transparent partition on top of more than one other partition. In other words, you cannot create a transparent partition target from multiple sources because each cell in a database must be retrieved from only one location —either the local disk or a remote disk.

Figure 53 Invalid Transparent Partition



- Carefully consider any formulas you assign to members in the data source and data target.

Advantages of Transparent Partitions

Transparent partitions can solve many database problems, but transparent partitions are not always the ideal partition type.

- You need less disk space, because you are storing the data in one database.
- The data accessed from the data target is always the latest version.
- When the user updates the data at the data source, Essbase makes those changes at the data target.
- Individual databases are smaller, so they can be calculated more quickly.
- The distribution of the data is invisible to the end user and the end user's tools.
- You can load the data from either the data source or data target.
- You can enable write-back functionality for aggregate storage databases by creating a transparent partition between an aggregate storage database as the source and a block storage database as the target.

  See "Using a Transparent Partition to Enable Write-Back for Aggregate Storage Databases" on page 936.

Disadvantages of Transparent Partitions

If these disadvantages are too serious, consider using replicated or linked partitions instead.

- Transparent partitions increase network activity, because Essbase transfers the data at the data source across the network to the data target. Increased network activity results in slower retrieval times for users.
- Because more users are accessing the data source, retrieval time may be slower.
- If the data source fails, users at both the data source and the data target are affected. Therefore, the network and data source must be available whenever users at the data source or data target need them.
- You can perform some administrative operations only on local data. For example, if you archive the data target, Essbase archives only the data target and not the data source. The following administrative operations work only on local data in block storage databases:
  - CLEARDATA calculation command
  - DATACOPY calculation command
  - EXPORT command
  - VALIDATE command
  - BEGINARCHIVE and ENDARCHIVE commands
  - Restructure operations in Administration Services
- When you perform a calculation on a transparent partition, Essbase performs the calculation using the current values of the local data and transparent dependents. Essbase does not recalculate the values of transparent dependents, because the outlines for the data source and the data target may be so different that such a calculation is inaccurate. To calculate all

partitions, issue a CALC ALL command for each individual partition, and then perform a CALC ALL command at the top level using the new values for each partition.

Consider this example:

  - The data target outline contains a Market dimension with East, West, South, and Central members
  - The data source outline contains an East dimension with New York and New Jersey members

  If you tried to calculate the data target outline, you would assume that East was a level 0 member. In the data source, however, East is derived by adding New York and New Jersey. Any calculations at the data target, however, would not know this information and could not reflect changes made to New York and New Jersey in the data source. To perform an accurate calculation, therefore, calculate East in the data source and then calculate the data target.

- Formulas assigned to members in the data source may produce calculated results that are inconsistent with formulas or consolidations defined in the data target, and vice versa.

Performance Considerations for Transparent Partitions

To improve the performance of transparent partitions, consider the following guidelines when creating the partition:

- Partitioning along dense dimensions in a transparent partition can greatly slow performance, because dense dimensions are used to determine the structure and contents of data blocks. If a database is partitioned only along a dense dimension at the target, Essbase must compose data blocks by performing network calls for the remote data in the transparent partition, in addition to the disk I/O for the local portion of the block.

  To improve performance, consider including one or more sparse dimensions in the area definition so that the number of blocks required is limited to combinations with the sparse members.

- Basing transparent partitions on the attribute values of a dimension can increase retrieval time, because attributes are associated with sparse dimensions. In such cases, partitioning at a level higher than the level that is associated with attributes improves retrieval time. For example, in the Product dimension of the Sample.Basic database, if children 100-10, 200-10, and 300-10 (level 0) are associated with attributes, then partition their parents 100, 200, and 300 (level 1) for better retrieval performance.
- Loading data into the data source from the data target can greatly slow performance. If possible, load data into the data source locally.
- Retrieval time is slower because users access the data over the network.
- When a transparent partition is the target, consider using these configuration settings:
  - For requests sent from a data source to a transparent partition target (whether a block storage or aggregate storage database), you can log transaction response times using the ENABLE_DIAG_TRANSPARENT_PARTITION configuration setting in the

    `essbase.cfg` file. Logging these messages is helpful when troubleshooting response times that are too slow.

  ❍ When the transparent partition target is an aggregate storage database, you can specify the maximum size of the request grid and the response grid, using the MAX_REQUEST_GRID_SIZE and MAX_RESPONSE_GRID_SIZE configuration settings.

- Partitioning base dimensions can greatly slow performance.
- For calculation-related performance considerations, see "Performance Considerations for Transparent Partition Calculations" on page 228.

Calculating Transparent Partitions

When you perform a calculation on a transparent partition, Essbase performs the calculation using the current values of the local data and transparent dependents. When calculating local data that depends on remote data, Essbase performs a bottom-up calculation. The bottom-up calculation can be done only if the calculator cache on the target database is used properly. See "Using Bottom-Up Calculation" on page 879.

Increasing the memory assigned to the calculator cache greatly improves calculation performance with transparent partitions. When a calculation is started, a message in the application log indicates whether the calculator cache is enabled or disabled on the target database. Using the calculator cache on the target database reduces the number of blocks that are requested from the data source during calculation. Reducing the blocks requested, in turn, reduces the network traffic that is generated by transferring blocks across the network. See "Sizing the Calculator Cache" on page 830.

Performance Considerations for Transparent Partition Calculations

Calculating data on the data target can greatly slow performance when the data target must retrieve each dependent data block across the network, and then perform the calculation.

Performance with transparent calculations also may slow if Essbase must perform a top-down calculation on any portion of the data target that contains top-down member formulas. When the data target does not contain top-down member formulas, Essbase can perform a bottom-up calculation on the data target, which is much faster.

When Essbase performs the calculation on the data source, it can always perform a bottom-up calculation. For a comparison of top-down and bottom-up calculations, see "Using Bottom-Up Calculation" on page 879.

Consider using these calculation alternatives:

- If you are absolutely sure that a target partition calculation script does not involve access to remote data, you can use the SET REMOTECALC OFF calculation command in the calculation script to stop retrieval efforts from the source partition. See the *Oracle Essbase Technical Reference*.

- Dynamic Calc or Dynamic Calc and Store members as parents of the transparent data so that the data is calculated on the fly when it is retrieved. This process reduces the batch processing time. Essbase performs the calculation only when users request it.
- A replicated layer between the low-level transparent data and high-level local data.

Consider these performance strategies:

- Keep the partition fully within the calculator cache area (see "Sizing the Calculator Cache" on page 830), which means that any sparse members in the partition definition must be contained within the calculator cache. For example, in the Sample.Basic database, if a partition definition includes @IDESC(East), all descendants of East must be within the calculator cache.
- Enable the calculator cache, and assign a sufficient amount of memory to it.
- Do not use complex formulas on any members that define the partition. For example, in Sample.Basic, assigning a complex formula to New York or New Jersey (both children of East) forces Essbase to use the top-down calculation method. See "Bottom-Up and Top-Down Calculation" on page 880.

Transparent Partitions and Member Formulas

If the data target and data source outlines are identical except for different member formulas, ensure that the partition definition produces the calculation results you want.

For example, suppose that the data source and data target outlines both contain a Market dimension with North and South members, and children of North and South. On the data target, Market is calculated from the data for the North and South members (and their children) on the data source. If any of these members on the data source contains member formulas, these formulas are calculated, affecting the calculated value of Market on the data target. These results may be different from how the Market member are calculated from the North and South members on the data target, where these formulas may not exist.

Ensure that any formulas you assign to members in the data source and data target produce the results you want.

Transparent Partitions and Port Usage

One port is used for every unique user and machine combination. If a user defines several transparent partitions on one server, using the same user name, then only one port is occupied.

In a transparent partition, when a user (user1) drills into an area in the target that accesses source data, user1 is using the user name declared in the partition definition (partition user) to access the data from the source database. This access causes the use of an additional port because different users (user1 and partition user) are connecting to the application.

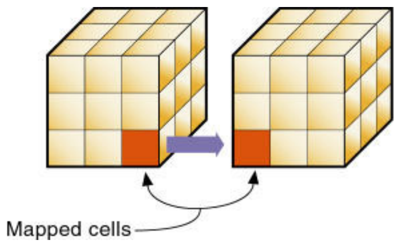
If a second user (user2) connects to the target database and drills down to access source data, user2 also uses the user name declared in the partition definition (partition user). Because the partition user is already connected to the source database, an additional port is not needed for the partition user, as long as user2 is accessing the same source database.

Linked Partitions

A linked partition connects two databases with a data cell. When you click the linked cell in the data target, you drill across to a second database—the data source—and view the data there. If you are using Spreadsheet Add-in, for example, a new sheet opens, displaying the dimensions in the second database. You can then drill down into these dimensions.

Unlike replicated or transparent partitions, linked partitions do not restrict you to viewing data in the same dimensionality as the target database. The database that you link to can contain different dimensions than the database from which you connected. With linked partitions, data is not physically transferred from the source to the target. Instead, a data cell or range of cells on the target provides a link point to a cell or range of cells on the source.

Figure 54 Linked Partition



To prevent users from seeing privileged data, establish security filters on the data source and the data target. See "Security for Partitioned Databases" on page 219.

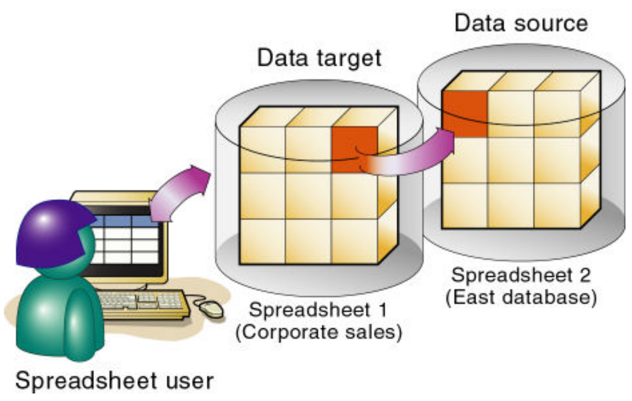
There are no performance considerations for linked partitions, beyond optimizing the performance of each linked database.

For example, if TBC grew into a large company, it might have several business units. Some data, such as profit and sales, exists in each business unit. TBC can store profit and sales in a centralized database so that the profit and sales for the entire company are available at a glance. The DBA can link business unit databases to the corporate database. See "Case Study 3: Linking Two Databases" on page 235.

A user in such a scenario can perform these tasks:

- View the general profit and sales at the corporate level in a spreadsheet at the data target.
- Drill across to individual business units, such as East (this action opens a new spreadsheet).
- Drill down in the new spreadsheet to more-detailed data.

Figure 55 Source and Target for Linked Partition



For linked partitions, the spreadsheet that the user first views is connected to the data target, and the spreadsheet that opens when the user drills across is connected to the data source. This setup is the opposite of replicated and transparent databases, in which users move from the data target to the data source.

Use a linked partition to connect databases with different dimensionality.

Advantages of Linked Partitions

- You can view data in a different context; that is, you can navigate between databases containing many dimensions.
- You need not keep the data source and data target outlines closely synchronized, because less of the outline is shared.
- A single data cell can allow the user to navigate to more than one database. For example, the Total Profit cell in the Accounting database can link to the Profit cells in the databases of each business unit.
- Performance may improve, because Essbase accesses the database directly, not through a data target.

Disadvantages of Linked Partitions

You must create an account for users on each database or use default access to the destination database (such as through a guest account). See "Drill Across and Linked Partitions" on page 231.

Drill Across and Linked Partitions

When a user clicks on a linked cell in a linked partition, a spreadsheet opens and displays the linked database. This process is called *drill across*. To facilitate drill-across access, you can use the following strategies:

- Create accounts for each user on each database. For example, if Mary accesses data in a Company database and an East database, create an account with the same login and password for Mary on the Company and East databases.

  See "Managing Users and Groups in Native Security Mode" on page 630.

- Create a default account that users can use when accessing target databases. For example, if users access data through a data source named Company and a data target named East, create a guest account for the East database with the appropriate permissions. Use the guest account login and password as the default login when creating the linked partition.

When a user drills across on data to a data target, Essbase logs the user into the data target using the following steps:

1. Checks whether the user has an account on the data target with the same name and password. If so, Essbase logs in the user using that account.
2. Checks whether you specified a default account on the data target when you created the partition. If you did, Essbase logs the user in using that account.
3. Opens a login window prompting the user to enter a new login and password. Once the user enters a valid login and password, Essbase logs the user in using that account.

Linked Partitions and Port Usage

When accessing a linked partition, Essbase tries to use the end user's (user1) login information to connect to the source database. If user1 does not have access to the source database, Essbase looks for the linked partition default user name and password. If these defaults are not specified, user1 is asked to enter login information to access the source database. Port usage varies depending on the number of user names being used to access the various source and target databases (and whether those databases are contained within the same or different servers).

Case Studies for Designing Partitioned Databases

The following sections describe examples of partitioning a database:

Case Study 1: Partitioning an Existing Database

Assume that TBC, the fictional soft drink company upon which the Sample.Basic database is based, started out with a centralized database. As the eastern region grew, however, this solution was no longer feasible. The networks to the eastern region could not handle the large data flow. Users were constantly waiting for data that they needed in order to make decisions. One day, the network went down, and users at the eastern region could not access the data.

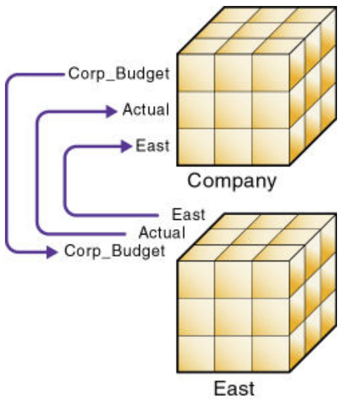
Everyone agreed that the eastern region needed to access its own data directly, without going through the company database. In addition, TBC decided to change where budgeting information was stored. The corporate budget stays at company headquarters, but the eastern region budget moves to the eastern region's database.

So, assume that TBC decided to ask you to partition their large centralized database into two smaller databases—Company and East.

This example is based on the Samppart sample application (which contains the Company database) and the Sampeast sample application (which contains the East database).

Figure 56 shows a subset of the partitioned databases. The arrows indicate flow from the data source to the data target. The Company database is the data source for the Corp_Budget member and the data target for the East and the East Actual members. The East database is the data source for its East and Actual members and the data target for the Corp_Budget member.

Figure 56 Data Flow from Data Source to Data Target



➤ To create a partition based on this example:

1 **Determine which data to partition.**

The Sample.Basic database contains five standard dimensions—Year, Measures, Product, Market, and Scenario.

- Partition the database along the East member of the Market dimension to give the eastern region more control over the contents of its database.
- Partition the database along the Actual and Corp_Budget members of the Scenario dimension.

2 **Choose the data source and the data target.**

- For Corp_Budget, use Company as source and East as Target, because the company owns the corporate budget—it is the source.
- For Eastern Region and Actual, East is the source and Company is the target, because the eastern region needs to update its market and actual information.

3 **Decide which type of partition to use.**

- For East, use transparent because the data target (Company) needs up-to-the-minute data.
- For Corp_Budget, use transparent because the data target (East) needs up-to-the-minute data.
- For East Actual, use replication because the data target (Company) does not need up-to-the-minute data.

4 **Create the partitioned databases by performing the following tasks.**

- Create the Sampeast application.
- Create the East database by cutting the Company outline and pasting it into the East outline. Then delete the extra members (South, West, and Central) and promote East.
- If necessary, edit existing data sources, rules files, calculation scripts, report scripts, and outlines.
- Create the partitions.
- Load data into the new partitions.

After the corporate database is partitioned, users and DBAs see the following benefits:

- Faster response times, because they are competing with fewer users for the data and they are accessing the data locally
- Easier maintenance, because DBAs can control the downtime of their local databases
- Access to more data, because users can connect to both the eastern and corporate budgets
- Higher-quality data, because the corporate budget and eastern budget are now synchronized —they use the same data.

Case Study 2: Connecting Existing Related Databases

Assume that TBC has several databases, such as Inventory, Payroll, Marketing, and Sales. Users viewing the Sample.Basic database want to share data with and navigate to those other databases. You, the DBA, want to synchronize related data. It is impractical to combine all of the databases into one database, for the following reasons:

- So many users access it that performance is slow.
- You cannot find downtime to administer the database.
- No one has control over his own data, because it is centrally managed.
- The database is very sparse, because so much of the data is unrelated.

By connecting the databases instead, you can:

- Leverage work that has already been completed
- Synchronize the data

Note:

This example is not included with Essbase.

➤ To connect multiple databases:

1 **Determine which data to connect. First, connect the Inventory database.**

- Replicate the Opening_Inventory and Ending_Inventory members from the Measures dimension of the Inventory database into the Measures dimension of the Sample.Basic database.

- Do not replicate the Number_On_Hand, Number_Shipped, and Number_Returned members in the Measures dimension of the Inventory database to the Sample.Basic database.
- Add a link to the Inventory database so that users can view these more-detailed measures, if necessary.
- Create a partition containing data from the Payroll, Marketing, and Sales databases in the Sample.Basic database.

2 **Choose the data source and data target.**

In the case of the Opening_Inventory and Ending_Inventory members, the Inventory database is the data source and the Sample.Basic database is the data target.

3 **Decide which type of partition to use.**

Use a replicated partition for the Opening_Inventory and Ending_Inventory members because the network connection is slow.

4 **Connect the Payroll, Marketing, and Sales databases. Perform steps step 1 through step 3 for each database.**

5 **Create the partitioned databases by performing the following tasks:**

- Edit existing data sources, rules files, calculation scripts, report scripts, and outlines
- Create the partitions
- If necessary, load data into the new partitions

Now that the Sample.Basic database is partitioned, users and DBAs see the following benefits:

- Easier maintenance, because DBAs can control the downtime of local databases.
- Access to more data; users can link to new databases.
- Higher-quality data, because the databases are now synchronized (they use the same data).

Case Study 3: Linking Two Databases

Assume that TBC has two main databases—Sample.Basic and TBC.Demo. Both databases have similar outlines, but TBC.Demo has two additional dimensions:

- Channel, which describes where a product is sold
- Package, which describes how the product is packaged

The DBA for Sample.Basic notices that more users are requesting that she add channel information to Sample.Basic. But, because she does not own the data for channel information, she is reluctant to do so. Instead, she decides to allow her users to link to the TBC.Demo database, which already contains this information.

Note:

This example is not shipped with Essbase.

➤ To link two databases:

1 **Determine which data to link.**

The DBA decides to link the Product dimension of Sample.Basic to the Product dimension of TBC.Demo.

Users can then drill across to TBC.Demo and view the Channel and Package information.

2 **Choose the data source and the data target.**

Because users start at the Sample.Basic database, it is considered the data target. Because users move to TBC.Demo, it is considered the data source.

Note:

This setup is the opposite of replicated and transparent databases, in which users move from the data target to the data source.

3 **Decide which type of partition to use.**

Use a linked partition because the databases have different dimensionality.

4 **Create the partition:**

- Establish a link from the Product member of Sample.Basic to the Product dimension of TBC.Demo. Remember to map the extra dimensions from TBC.Demo—Channel and Product—to void in Sample.Basic. See "Mapping Data Cubes with Extra Dimensions" on page 241.
- Set up a guest account on TBC.Demo that gives the users who connect from Sample.Basic permissions to access the Channel and Package dimensions. For a general discussion on creating accounts, see "Granting Permissions to Users and Groups in Native Security Mode" on page 628. To assign accounts to linked partitions, see "Choosing a Partition Type" on page 237 and "Choosing a Partition Type" on page 237.

After the databases are linked, users and DBAs see the following benefits:

- Users have access to more data.
- The DBA for Sample.Basic need not maintain TBC.Demo; she needs only to check the link periodically to make sure that it still works.

15 Creating and Maintaining Partitions

In This Chapter

Process for Creating Partitions

When you build a partition, each database in the partition uses a partition definition file to record all information about the partition, such as its data source, data target, and the areas to share. Partition creation requires Database Manager permissions or higher.

After you have created a partition, load and calculate the database that contains the partition. Loading and calculating the partition may require you to change existing rules files and calculation scripts. See Chapter 16, “Understanding Data Loading and Dimension Building” and Chapter 21, “Calculating Essbase Databases.”

Choosing a Partition Type

Decide which type of partition to create:

- Replicated
- Transparent

- Linked

See "Partition Types" on page 211.

Setting up the Data Source and the Data Target

Define the data source and data target, including specifying the source application and database, the target application and database, and the location of each. For a list of supported block and aggregate storage data source and data target combinations for each partition type, see "Data Sources and Data Targets" on page 213.

➤ To set up the data source and the data target:

1. **Specify the names of the application and database for the data source and data target.**

   See "Specifying Connection Information for Partitions" in the *Oracle Essbase Administration Services Online Help*.

2. **Specify the locations of the Essbase Servers on which the data source and data target reside.**

   See "Specifying Connection Information for Partitions" in the *Oracle Essbase Administration Services Online Help*.

   If you want to use network aliases for the data source or data target names, ensure that the aliases are propagated to all computers on your system. Otherwise, use the full server name.

   To propagate an alias to all the computers on your system, edit the `hosts` file (you need root or administrative privileges) for the operating system you are using:

   - Windows: `%WINDIR%/system32/drivers/etc/hosts`
   - UNIX: `/etc/hosts`

   In the `hosts` file, add an entry using the following syntax:

   ```
   IP_address hostname.domainname alias [alias]
   ```

   The following example specifies one alias:

   ```
   172.234.23.1 myhost.mydomain abcdefg.hijk.123
   ```

   This example specifies multiple aliases:

   ```
   172.234.23.1 myhost.mydomain abcdefg.hijk.123 lmnopqrs.tuvw.456
   ```

   Note:

   Do not use localhost as an alias to specify source and target server names.

3. **Optional: Enter a note to describe the data source or data target.**

   See "Specifying Connection Information for Partitions" in the *Oracle Essbase Administration Services Online Help*.

4. **Optional: Specify the outline to which you can make changes.**

By default, all changes made on the data source outline overwrite the data target outline when you synchronize the outlines. You can, however, specify that changes made to the data target outline overwrite the data source outline when you synchronize the outlines.

See “Synchronizing Outlines” on page 248.

Setting the User Name and Password

You must specify a user name and password for Essbase to use when it communicates between the data source and the data target. The user name and password must be identical on the data source and the data target. Essbase uses this user name and password to:

- Transfer data between the data source and the data target for replicated and transparent partitions. Local security filters apply to prevent end users from seeing privileged data.
- Synchronize database outlines for all partition types.

See “Security for Partitioned Databases” on page 219.

➤ To set the user name and password for the data source and the data target, see “Specifying Connection Information for Partitions” in the *Oracle Essbase Administration Services Online Help*.

Defining a Partition Area

You can define or edit the areas of the data source to share with the data target in a partition. An area is a subcube within a database and a partition comprises one or more areas. For example, an area could be all Measures at the lowest level for Actual data in the Eastern region.

When you define a replicated area, ensure that the data source and data target contain the same number of cells. The two partitions must have the same shape. For example, if the area in the data source covers 18 cells, the data target should contain an area covering 18 cells into which to put those values. The cell count does not include the cells of attribute dimensions.

You can use substitution variables in a partition area definition, which provides flexibility in sharing different data at different times. See “Substitution Variables in Partition Definitions” on page 215.

Note:

Use member names instead of their aliases to create area definitions. Although Essbase validates the aliases, the partitions will not work.

➤ To define a partition area, see “Defining Areas in Partitions” in the *Oracle Essbase Administration Services Online Help*.

Mapping Members

To create a partition, Essbase must be able to map all shared data source members to data target members. Oracle recommends that data source member names and data target member names are the same to reduce maintenance requirements for the partition, especially when the partition is based on member attributes.

If the data source and data target contain the same number of members and use the same member names, Essbase automatically maps the members. You need only validate, save, and test the partitions. If Essbase cannot map automatically, you must map manually.

Map data source members to data target members in any of the following ways:

- Enter or select member names manually. (When you type a duplicate member name, type the qualified member name and enclose it in double quotation marks; for example, `"[State].[New York]"`
- Import the member mappings from an external data file.
- Create area-specific mappings.

Note:

You can use substitution variables for member names in mapping specifications. See “Substitution Variables in Partition Definitions” on page 215.

To map members, see “Defining Global Mappings in Partitions” in the *Oracle Essbase Administration Services Online Help*.

Mapping Members with Different Names

If the data source outline and data target outline contain different members, or if the members have different names in each outline, you must map the data source members to the data target members. In the following example, the first two member names are identical, but the third member name is different:

```
Source      Target

Product   Product
   Cola      Cola

Year      Year
   1998      1998
```

Source	Target

```
Market   Market
   East     East_Region
```

Because you know that East in the data source corresponds to East_Region in the data target, map East to East_Region. Then, all references to East_Region in the data target point to East in the data source. For example, if the data value for Cola, 1998, East is 15 in the data source, the data value for Cola, 1998, East_Region is 15 in the data target.

Mapping Data Cubes with Extra Dimensions

The number of dimensions in the data source and data target may vary. The following example illustrates a case where there are more dimensions in the data source outline than in the data target outline:

Source	Target

```
Product  Product
   Cola     Cola

Market   Market
   East     East

Year
   1999
   1998
   1997
```

You can map member 1997 of the Year dimension to Void in the data target, but first define the areas of the data source to share with the data target:

Source	Target
`@DESCENDANTS(Market), 1997`	`@DESCENDANTS(Market)`

You can then map the data source member to Void in the data target:

Source	Target
`1997`	`Void`

“Void” is displayed automatically; manually entering “Void” may cause errors.

If you do not include at least one member from the extra dimension in the area definition, you will receive an error message when you attempt to validate the partition.

Note:

When you map a member from an extra dimension, the partition results reflect data only for the mapped member. In the above example, the Year dimension contains three members: 1999, 1998, and 1997. If you map member 1997 from the data source to the data target, the partition results reflect Product and Market data only for 1997. Product and Market data for 1998 and 1999 will not be extracted.

The following example illustrates a case where the data target includes more dimensions than the data source:

```
Source     Target

Product    Product
   Cola       Cola

           Market
              East

Year       Year
   1997       1997
```

In such cases, first define the shared areas of the data source and the data target:

```
Source                   Target

@IDESCENDANTS(Product)   @IDESCENDANTS(Product), East
```

You can then map member East from the Market dimension of the data target to Void in the data source:

```
Source  Target

Void    East
```

If member East from the Market dimension in the data target is not included in the target areas definition, you will receive an error message when you attempt to validate the partition.

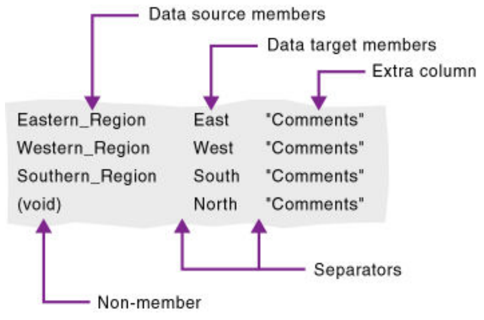
Mapping Shared Members

When you create a replicated or transparent partition using a shared member, use the actual member names in the mapping. Essbase maps the actual member from the data source.

Importing Member Mappings

You can import member mappings from a text file. Mapping files must have the `.txt` extension. A sample member file must contain all of the following (except extra columns):

Figure 57   Member Mapping Import File



- Data target members column—lists the member names in the data target. Member names containing spaces must be in quotes.
- Data source members column—lists the member names in the data source. Member names containing spaces must be in quotation marks.
- Nonmember column—missing members. Use it to map an extra member in the data source to Void in the data target or to map an extra member in the data target to Void in the data source.
- Separators—tabs or spaces to separate columns.
- Extra column—the file can contain extra columns that do not contain member names.

➤ To import member mappings, see “Importing Member Mappings for Partitions” in the *Oracle Essbase Administration Services Online Help*.

Mapping Attributes Associated with Members

You must accurately map attribute dimensions and members from the data source to the data target to ensure that the partition is valid.

Note:

You cannot map members of attributes dimension in replicated partitions (see “Rules for Replicated Partitions” on page 221). You can, however, map attributes in transparent and linked partitions (see “Attributes in Partitions” on page 215).

In the following example, the outline for the data source contains a Product dimension with a member 100 (Cola). Children 100-10 and 100-20 are associated with member TRUE of the Caffeinated attribute dimension, and child 100-30 is associated with member FALSE of the Caffeinated attribute dimension.

The data target outline has a Product dimension with a member 200 (Cola). Children 200-10 and 200-20 are associated with member Yes of the With_Caffeine attribute dimension, and child 200-30 is associated with No of the With_Caffeine attribute dimension.

First define the areas to be shared from the data source to the data target:

```
Source                Target

@DESCENDANTS(100)   @DESCENDANTS(200)
@DESCENDANTS(East)  @DESCENDANTS(East)
```

Then map the attributes:

```
Source              Target

100-10              200-10
100-20              200-20
100-30              200-30
Caffeinated         With Caffeine
Caffeinated_True    With_Caffeine_True
Caffeinated_False   With_Caffeine_False
```

If you map attribute Caffeinated_True to attribute With_Caffeine_No, you receive an error message during validation. You must associate caffeinated cola from the data source to caffeinated cola in the data target.

An attribute dimension or an attribute member can exist in the outline of the data source but not in the outline of the data target, or in the outline of the data target but not in the outline for the data source. For example:

```
Source          Target

Caffeinated
   True
   False
```

In such cases, you have the following choices:

- Create the Caffeinated attribute dimension and its members in the outline of the data target and associate them with the Product dimension. You can then map the attributes from the data source to the data target.
- Map the Caffeinated attribute dimension in the data source to Void in the data target.

For a comprehensive discussion of attributes, see Chapter 10, “Working with Attributes.” For a general discussion of attributes in partitions, see “Attributes in Partitions” on page 215.

Creating Advanced Area-Specific Mappings

If you can map all of the members in your data source to their counterparts in the data target using standard member mapping, you need not perform advanced area-specific mapping.

If, however, you need to control how Essbase maps members at a more granular level, you may need to use area-specific mapping, which maps members in one area to members in another area only in the context of a particular area map.

Use area-to-area mapping to do the following:

- Map data differently depending on where it is coming from.
- Map more than one member in the data source to a single member in the data target.

Because Essbase cannot determine how to map multiple members in the data source to a single member in the data target, you must logically determine how to divide your data until you can apply one mapping rule to that subset of the data. Then use that rule in the context of area-specific mapping to map the members.

➤ To create area-specific mappings, see "Defining Area-Specific Member Mappings in Partitions (Optional)" in the *Oracle Essbase Administration Services Online Help*.

Example 1: Advanced Area-Specific Mapping

The data source and data target contain the following dimensions and members:

```
Source     Target

Product    Product
   Cola       Cola

Market     Market
   East       East

Year       Year
   1998       1998
   1999       1999

           Scenario
              Actual
              Budget
```

The data source does not have a Scenario dimension. Instead, it assumes that past data is actual data and future data is forecast, or budget, data.

You know that 1998 in the data source should correspond to 1998, Actual in the data target and 1999 in the data source should correspond to 1999, Budget in the data target. So, for example, if the data value for Cola, East, 1998 in the data source is 15, the data value for Cola, East, 1998, Actual in the data target should be 15.

Because mapping works on members, not member combinations, you cannot simply map 1998 to 1998, Actual. Define the area (1998 and 1998, Actual) and then create area-specific mapping rules for that area.

Because the data source does not have Actual and Budget members, you also must map these members to Void in the data target.

Example 2: Advanced Area-Specific Mapping

You also can use advanced area-specific mapping if the data source and data target are structured very differently but contain the same kind of information.

This strategy works, for example, if your data source and data target contain the following dimensions and members:

```
Source      Target

Market      Customer_Planning
   NY          NY_Actual
   CA          NY_Budget
               CA_Actual
               CA_Budget

Scenario
   Actual
   Budget
```

You know that NY and Actual in the data source should correspond to NY_Actual in the data target and NY and Budget in the data source should correspond to NY_Budget in the data target. So, for example, if the data value for NY, Budget in the data source is 28, the data value for NY_Budget in the data target should be 28.

Because mapping works on members, not member combinations, you cannot simply map NY, Actual to NY_Actual. Define the area (NY and Actual, and NY_Actual) and then create area-specific mapping rules for that area.

Because the data target does not have NY and CA members, you must also map these members to Void in the data target so that the dimensionality is complete when going from the data source to the data target.

Validating Partitions

When you create a partition, validate it to ensure its accuracy before you use it. Database Manager permissions or higher are required. After you validate, save the partition definition. If necessary, you can edit an existing partition.

When Essbase validates a partition definition, it checks on the Essbase Server for the data source and the data target to ensure that:

- The area definition is valid (contains no syntax errors).
- The specified data source members are valid and map to valid members in the data target.
- All connection information is correct; that is, the server names, database names, application names, user names, and password information.
- For linked partitions, the default user name and password that you provide are correct.
- For replicated and transparent partitions, a replication target does not overlap with a replication target; a replication target does not overlap with a transparent target; and a transparent target does not overlap with a transparent target.
- For replicated and transparent partitions, the cell count for the partition is the same on the data source and the data target.
- For replicated and transparent partitions, the area dimensionality matches the data source and the data target.

- You must validate a transparent partition that is based on attribute values to ensure that the results are complete. Essbase does not display an error message when results are incomplete.

After you validate, save the partition; the partition definition is saved to two `.ddb` files, on the data source server and the data target server.

➤ To validate a partition, use a tool:

Tool	Topic	Location
Administration Services	Validating Partitions	*Oracle Essbase Administration Services Online Help*
ESSCMD	VALIDATEPARTITIONDEFFILE	*Oracle Essbase Technical Reference*
MaxL	**create partition**	*Oracle Essbase Technical Reference*

Saving Partitions

After you validate the partition definition, you can save the partition definition to any of the following locations:

- To both the data source server and the data target server. The partition definition is stored in two `.ddb` files.
- To a client machine. The partition definition is stored in a `.ddb` file.

Note:

Although you can save a partition with mapping errors, operations using that partition will fail until the mapping errors are fixed.

➤ To save a partition definition, see “Saving Partitions” in the *Oracle Essbase Administration Services Online Help*.

Process for Maintaining Partitions

The following sections describe how to maintain partitions.

Testing Partitions

To test a partition:

- View data targets using the Spreadsheet Add-in or another tool to ensure that the user sees the correct data.
- When testing a linked partition, ensure that Essbase links you to the expected database and that the default user name and password work correctly.

Synchronizing Outlines

When you partition a database, Essbase must be able to map each dimension and member in the data source outline to the appropriate dimension and member in the data target outline. After you map the two outlines to each other, Essbase can make the data in the data source available from the data target, as long as the outlines are synchronized and the partition definitions are up-to-date.

If you make changes to one outline, the two outlines are no longer synchronized. Although Essbase makes whatever changes it can to replicated and transparent partitions when the outlines are not synchronized, Essbase may not be able to make the data in the data source available in the data target.

Essbase tracks changes that you make to block storage outlines and provides tools to keep your block storage outlines synchronized.

Note:

Essbase does not enable automatic synchronization of aggregate storage outlines. You must manually make the same changes to the source and target outlines.

Setting the Source Outline and the Target Outline

Before you can synchronize block storage outlines, you must determine which outline is the source outline and which is the target outline.

- The source outline is the outline from which outline changes are taken.
- The target outline is the outline to which outline changes are applied.

By default, the source outline is from the same database as the data source; that is, outline and data changes flow in the same direction. For example, if the East database is the data source and the Company database is the data target, the default source outline is East.

You can also use the data target outline as the source outline. Consider this method if the structure of the outline (its dimensions, members, and properties) is maintained centrally at a corporate level, while the data values in the outline are maintained at the regional level (for example, East). Administrators can make changes in the Company outline and apply those changes to each regional outline when the outline is synchronized.

- If you make changes to the shared area in the source outline, you can propagate them to the target outline when you synchronize the outlines.
- If you make changes to the target outline, those changes cannot be propagated back to the source outline when you synchronize the outlines. To move these changes up to the source outline, make those changes in Outline Editor. See Chapter 7, “Creating and Changing Database Outlines.”

Essbase updates as many changes as possible to the target outline. If Essbase cannot apply all changes, a warning message prompts you to see the application log for details. Messages that pertain to outline synchronization are prefixed with OUTLINE SYNC. See “Viewing the Essbase Server and Application Logs” on page 742.

➤ To set the source outline, see "Setting up the Data Source and the Data Target" on page 238.

Performing Block Storage Outline Synchronization

➤ To synchronize block storage outlines, use a tool:

Tool	Topic	Location
Administration Services	Synchronizing Outlines	*Oracle Essbase Administration Services Online Help*
MaxL	**refresh outline**	*Oracle Essbase Technical Reference*
ESSCMD	GETPARTITIONOTLCHANGES APPLYOTLCHANGEFILE RESETOTLCHANGETIME PURGEOTLCHANGEFILE	*Oracle Essbase Technical Reference*

Note:

For synchronizing non-Unicode-mode outlines with multibyte characters, you can use only non-Unicode clients such as ESSCMD or MaxL statements executed through the MaxL Shell.

Note:

Outline synchronization cannot be performed on an outline containing a Dynamic Calc member that has many (approximately 100 or more) children.

Tracking Changes

The following table describes the process for changing the source outline and synchronizing the target outline with the source outline:

Action You Take	Action Essbase Takes
Make changes to the source outline	1. Records the changes in a change log named `ess`*`xxxxx`*`.chg`, where *`xxxxx`* is the number of the partition. If you have more than one partition on a source outline, Essbase creates a change log for each partition. 2. Creates or updates the outline change timestamp for that partition in the partition definition (`.ddb`) file. Each partition defined against the source outline has a separate timestamp in the `.ddb` file.
Pull changes from the outline source	1. Compares the last updated timestamp in the target outline `.ddb` file to the last updated timestamp in the source outline backup (`.dbb`) file. Essbase updates the target timestamp when it finishes synchronizing the outlines using

Action You Take	Action Essbase Takes
	the last updated time on the source outline, even if the two outlines are on servers in different time zones. 2. If the source outline has changed since the last synchronization, Essbase retrieves those changes from the source outline change log and places them in the target outline change log. The change logs may have different names on the source outline and the target outline.
Select the changes to apply to the target outline	1. Applies the changes to the target outline. 2. Updates the timestamp in the target outline's `.ddb` file, using the time from the source outline.

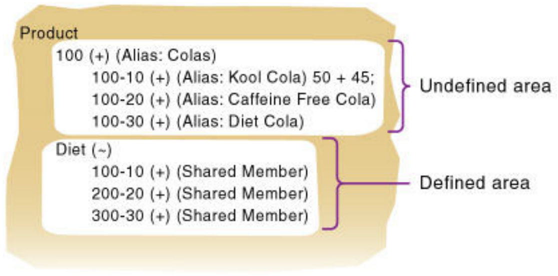
Caution!

If you choose not to apply some changes, you cannot apply those changes later.

Updating Shared Members During Outline Synchronization

An actual member and its shared members in the source outline are propagated to the target outline if at least one actual or shared member is defined in the partition area. As illustrated in Figure 58, the partition definition is @IDESC("Diet"). The parent 100 and its children (100-10, 100-20, 100-30) are not defined in the partition area. The parent Diet and its children (100-10, 100-20, 100-30) are defined in the partition area. The children of Diet are shared members of the actual members.

Figure 58 Shared Members and Outline Synchronization


Product
100 (+) (Alias: Colas)
100-10 (+) (Alias: Kool Cola) 50 + 45;
100-20 (+) (Alias: Caffeine Free Cola)
100-30 (+) (Alias: Diet Cola)
Diet (~)
100-10 (+) (Shared Member)
200-20 (+) (Shared Member)
300-30 (+) (Shared Member)
Undefined area
Defined area

If you make a change to an actual member in the undefined partition area, such as adding an alias to the 100-10 actual member, that change is propagated to the target outline because it is associated with a shared member in the defined partition area.

The reverse is also true. If a shared member is not in the partition area and its actual member is, a change to the shared member in the undefined area is propagated to the target outline.

Any change made to a member that does not have at least one actual member (or shared member) in the defined partition area is not propagated to the target outline. For example, in Figure 58, a change to the parent 100 is not propagated to the target outline because it is in the undefined partition area and does not have an associated shared member in the defined partition area.

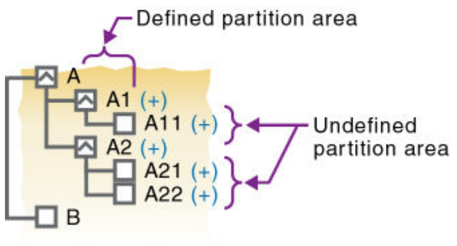
If a shared member is included in the partition area, it is recommended to include its parent. In the above example, the parent Diet is included in the outline because its children are shared members and in the defined partition area.

Implied shared members are treated the same as shared members during outline synchronization. Actual members and their implied shared members in the source outline are propagated to the target outline if at least one actual or implied shared member is defined in the partition definition.

Using the partition definition as @CHILD("A") in the example in Figure 59, A1 and A2 are in the defined partition area, and A11, A21, and A22 are in the undefined partition area. Although A11 (implied shared member) is in the undefined partition area, a change to A11 is propagated to the target outline because its parent, A1, is in the defined partition area. The change to the children A21 and A22 is not propagated to the target outline because these members are not defined in the partition area and are not associated with a member that is in the defined partition area.

The reverse is true again. If A1 is not defined in the partition area and its implied shared member is, any change to A1 is propagated to the target outline.

Figure 59 Implied Shared Members and Outline Synchronization



Populating or Updating Replicated Partitions

The administrator should regularly update data in a replicated partition. How frequently you update replicated partitions depends on user requirements for up-to-the-minute data. Essbase keeps track of when the data source was last changed and when the data target was last updated so that you can determine when to update replicated partitions. This information is saved at the data source. The administrator of either the data source site or data target site can be responsible for replicating data.

Essbase also tracks which cells in a partition are changed:

- Faster—Only the cells that have changed since the last replication
- Slower—All cells

  The slower update is useful under certain conditions; for example, updating all cells to recover lost data at the target.

Follow these guidelines:

- Unless you update all cells, replication does not update target data when the source data has not changed since the last replication.

- By default, Essbase replicates #MISSING cells. If you do not want to replicate #MISSING cells, you can use the DISABLEREPLMISSINGDATA configuration setting in the `essbase.cfg` file. See the *Oracle Essbase Technical Reference*.
- If you deleted data blocks on the data source, Essbase updates all data cells at the data target, even if you choose to update only changed cells. You can delete data blocks at the data source using any of these methods:
  - Using the CLEARDATA command in a calculation script
  - Using “Clear combinations” in your rules file during a data load
  - Issuing CLEAR UPPER, CLEAR INPUT or RESETDB commands in Administration Services;
  - Restructuring the database keeping only level 0 or input data
  - Deleting sparse members

You can replicate:

- All data targets connected to a data source.

  For example, if you replicate all data targets connected to the Sampeast.East database, Essbase updates the Budget, Actual, Variance, and Variance % members in the Samppart.Company database:
- From all data sources connected to a data target.

  For example, if you replicate from all data sources connected to the Samppart.Company database, Essbase pulls the Budget, Actual, Variance, and Variance % members from the Sampeast.East database and updates them in the Samppart.Company database.

➤ To update a replicated partition, use a tool:

Tool	**Topic**	**Location**
Administration Services	Replicating Data	*Oracle Essbase Administration Services Online Help*
MaxL	**refresh replicated partition**	*Oracle Essbase Technical Reference*
ESSCMD	GETUPDATEDREPLCELLS GETALLREPLCELLS PUTUPDATEDREPLCELLS PUTALLREPLCELLS	*Oracle Essbase Technical Reference*

Editing and Deleting Partitions

When you edit a partition, you use the same interface as for creating the partition.

When you delete a partition, Essbase deletes the partition definition from the `.ddb` file on the data source and data target servers.

➤ To edit or delete a partition, see “Opening the Create or Edit Partition Window” and “Deleting Partitions” in the *Oracle Essbase Administration Services Online Help*.

Viewing Partition Information

➤ To view information about a partition, use a tool:

Tool	Topic	Location
Administration Services	Opening the Create or Edit Partition Window	*Oracle Essbase Administration Services Online Help*
MaxL	**display partition**	*Oracle Essbase Technical Reference*
ESSCMD	PRINTPARTITIONDEFFILE	*Oracle Essbase Technical Reference*

Troubleshooting Partitions

The following table lists common problems that you may encounter when using partitions.

Symptom	Possible Cause	Solution
When replicating to multiple data targets, some are not replicated.	The connection between the data source and one of the data targets was lost during the replication operation.	Retry the replication operation. If one database is unavailable, replicate into only the available databases.
Not all information arrived at the data target.	The data source and the data target outlines are no longer mappable.	Synchronize outlines for the data source and the data target and try again.
A new or recently changed partition is validated and saved but does not function.	The partition may have a circular dependency. If database A is the source for database B, database B cannot be source for database A for the same slice.	Edit the partition definition to remove the circular dependency.
You keep running out of ports.	Partitions connect to other databases using ports.	Purchase more ports.
When you try to access a partition, you cannot connect to it.	Someone has deleted, renamed, or moved the application containing the database to which you are trying to connect.	Edit the partition having problems to specify the new application name or location.
Partitioned databases no longer can connect to each other.	Your host names may not match. Did you use the `hosts` file to provide aliases to your local machine? See “Setting up the Data Source and the Data Target” on page 238.	Ensure that the host names are synchronized between the servers.
Essbase overwrites user edits.	Users are changing data at a replicated partition that you overwrite each time you update the partition.	Set the partition to disallow user updates, or explain to users why their data disappears.

Symptom	Possible Cause	Solution
Administration Services does not reflect outline changes; that is, it lists the outlines as being in sync although one outline has changed.	● Was the target outline changed, but not the source outline? Essbase propagates changes only to the source outline. ● Does the outline change affect a defined partition? Essbase does not propagate changes to the outline that do not affect any partitions.	Examine the partition definition.
Data is confusing.	Your partition may not be set up correctly.	Check your partition to ensure that you are partitioning the data that you need.
Moved members or dimensions in the source outline are not reflected properly in a synchronized target outline.	Moving a dimension past a dimension that is not in the partition may not get propagated to the target during outline synchronization. Also, moving a member past a member that is not in the partition may not get propagated.	Structure your outline so that members and dimensions are not moved across partitions. If doing so is not possible, change the target outline to reflect the source outline moved members or dimensions.

Part III

Building Dimensions and Loading Data

In Building Dimensions and Loading Data:

16 Understanding Data Loading and Dimension Building

In This Chapter

The information in this chapter applies to block storage databases and aggregate storage database. As some rules file options and data source requirements vary for aggregate storage databases, also see "Preparing Aggregate Storage Databases" on page 954.

Introduction

An Essbase database contains dimensions, members, and data values.

- Loading data is the process of adding data values to an Essbase database from a data source, such as a Microsoft Excel spreadsheet or SQL database. If the data source is not perfectly formatted, you need a rules file to load the data values.
- Building dimensions is the process of loading dimensions and members to an Essbase database outline by using a data source and a rules file. You can also use Outline Editor to add dimensions and members manually.

Process for Data Loading and Dimension Building

To load data values or dimensions and members into an Essbase database, follow these steps:

1. Set up the data source.

   If you are not using a rules file, you must set up the data source outside Essbase. See "Data Sources" on page 258.

2. If necessary, set up the rules file.

   See "Rules Files" on page 263.

3. Perform the data load or dimension build.

   See Chapter 19, “Performing and Debugging Data Loads or Dimension Builds.”

Data Sources

Data sources contain the information that you want to load into the Essbase database. A data source can contain:

- Data values
- Information about members, such as member names, member aliases, formulas, and consolidation properties
- Generation and level names
- Currency name and category
- Data storage properties
- Attributes
- UDAs

The following sections describe the components of any kind of data source.

Supported Data Sources

Essbase supports the following types of data sources:

- Text files (flat files) from text backups or external sources
- SQL data sources
- Essbase export files (export files do not need a rules file to load)
- Microsoft Excel spreadsheet files
- Spreadsheet audit log files
- Oracle BI Enterprise Edition (OBIEE)

Note:

When using spreadsheet files to load data or build an outline in Essbase, the spreadsheet files must reside on a Windows computer, regardless of the tool you use. Spreadsheet files that reside on a UNIX computer or are transferred via FTP to a UNIX computer are not supported. If Essbase Administration Server is installed on a UNIX computer, data loads and dimension builds from client-side spreadsheet files are not supported.

To avoid rules file, data load, and dimension build errors, remove formatting in Microsoft Excel data source files; for example, in Excel, set color to “Automatic” and “No Fill,” and remove font settings such as bold and italic.

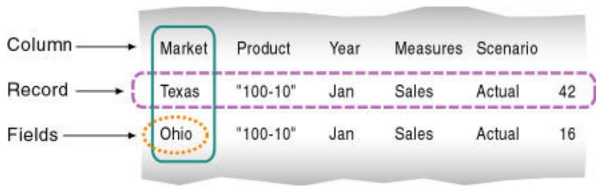
Items in a Data Source

As illustrated in Figure 60, a data source comprises records, fields, and field delimiters.

- A record is a structured row of related fields.
- A field is an individual value.
- A delimiter indicates that a field is complete and that the next character in the record starts another field.

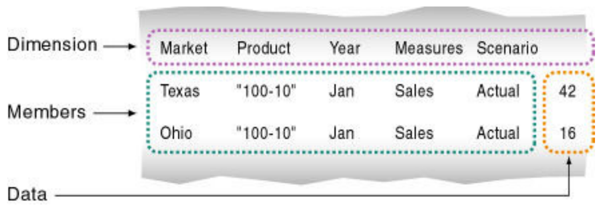
Essbase reads data sources starting at the top and proceeding from left to right.

Figure 60 Records and Fields


Column
Record
Fields
Market Product Year Measures Scenario
Texas "100-10" Jan Sales Actual 42
Ohio "100-10" Jan Sales Actual 16

As illustrated in Figure 61, data sources can contain dimension fields, member fields, member combination fields, and data fields.

Figure 61 Kinds of Fields


Dimension
Members
Data
Market Product Year Measures Scenario
Texas "100-10" Jan Sales Actual 42
Ohio "100-10" Jan Sales Actual 16

- Dimension fields identify the dimensions of the database, such as Market. Use dimension fields to tell Essbase the order of the dimensions in the data source. In Figure 61, for example, the dimension fields are Market, Product, Year, Measures, and Scenario. Fields in the Market column, such as Texas, are members of the Market dimension, and fields in the Product column, such as 100-10, are members of the Product dimension. Although you can set dimension fields in the data source, usually you define dimension fields in the rules file.
- Member fields identify the members or member combinations of the specified dimensions. Use member fields to tell Essbase to which members to map new data values, or which members to add to the outline. In Figure 61, for example, Texas, 100-10, Jan, Sales, and Actual are member fields.
- Data fields contain the numeric data values that are loaded into the intersections of the members of the database. Each data value must map to a dimension intersection. In Figure 61, for example, 42 is the data value that corresponds to the intersection of Texas, 100-10, Jan, Sales, and Actual.

  You can specify information in the header and in an individual record. In the following example, 100 is the data value that corresponds to the intersection of Jan, Actual, Cola, East, Sales, and 200 is the data value that corresponds to the intersection of Jan, Actual, Cola, West, Sales.

```
Jan, Actual
Cola  East  Sales  100
Cola  West  Sales  200
Cola  South  Sales  300
```

Data fields are used only for data loading; dimension builds ignore data fields.

The following sections describe each item in a data source.

Valid Dimension Fields

In a data load, if the data source does not identify every dimension in the Essbase database, the rules file must identify the missing dimensions. For example, the Sample.Basic database has a dimension for Year. If several data sources arrive with monthly numbers from different regions, the month itself may not be specified in the data sources. You must specify the month in the data source header or the rules file. See "Defining Header Records" on page 286.

A dimension field must contain a valid dimension name. If you are not performing a dimension build, the dimension must already exist in the database. If you are performing a dimension build, the dimension name can be new, but the new name must be specified in the rules file.

Valid Member Fields

A member field can contain the name of a valid member or an alias. In Figure 61 on page 259, for example, Texas and Ohio are valid members of the Market dimension. A blank member field inherits the member name from the previous record. Essbase must know how to map each member field of the data source to a member of the database.

To be valid, a member field must meet the following criteria:

- The member field must contain or inherit a valid member name or member property. See "Using the Data Source to Work with Member Properties" on page 275. If you are not performing a dimension build, the member must already exist in the outline. If you are performing a dimension build, the member can be new.
- Either the data source or the rules file must specify which dimension each member field maps to.
- A member field can map to a single member name, such as Jan (which is a member of the Year dimension), or to a member combination, such as Jan, Actual (which are members of the Year and Scenario dimensions).
- Member names that contain the same character as the file delimiter must be surrounded by double quotation marks. For example, if the data source is delimited by spaces, ensure that a member containing spaces, such as "New York," is enclosed by double quotation marks. If you are performing a data load without a rules file, member names containing some other characters also must be enclosed by quotation marks. See "Data Sources that Do Not Need a Rules File" on page 264.

When a rules file is not used, blank dimension and member fields are valid. When Essbase encounters a blank dimension or member field while loading data without a rules file, it uses the last dimension or member name encountered for that dimension or member column.

Note:

While it processes each record in a data source for a data load, Essbase does not check to ensure that a member specified in a member field belongs to the dimension specified for the dimension field. Essbase loads the data value to the data cell identified by the member combination in the record. In Figure 61 on page 259, for example, if the second record reversed Jan and Sales (Texas, '100-10', Sales, Jan, Actual, 42), Essbase would load 42 to the correct data cell. The exception is for fields in the rules file set as dimension reference method.

Valid Data Fields

If you are performing a dimension build, skip this section. Data fields are ignored during a dimension build.

Either the data source or the rules file must contain enough information for Essbase to determine where to put each data value. A data field contains the data value for its intersection in the database. In Figure 61 on page 259, for example, 42 is a data field. It is the dollar sales of 100-10 (Cola) in Texas in January.

In a data field, Essbase accepts numbers and their modifiers, with no spaces or separators between them, and the text strings #MI and #MISSING, as listed in Table 37.

Table 37 Valid Data Field Modifiers

Valid Modifiers	Examples
Currency symbols: ● Dollar $ ● Euro € ● Yen ¥	$12 is a valid value. $ 12 is not a valid value because there is a space between the dollar sign and the 12.
Parentheses around numbers to indicate a negative number	(12)
Minus sign before numbers. Minus signs after numbers are not valid.	-12
Decimal point	12.3
Large numbers with or without commas	1,345,218 and 1345218 are valid values.
#MI or #MISSING to represent missing or unknown values	See "Replacing an Empty Field with Text" on page 292.

If the data source contains a member field for every dimension and one field that contains data values, you must define the field that contains data values as a data field in the rules file. To read the following data source into the Sample.Basic database, for example, define the last field as a data field.

```
Jan     Cola     East     Sales     Actual     100
Feb     Cola     East     Sales     Actual     200
```

➤ To define a data field, see "Defining a Column as a Data Field" in the *Oracle Essbase Administration Services Online Help*.

If the data source contains blank fields for data values, replace them with #MI or #MISSING. If there is no value in the data field (or the value is #MISSING), Essbase does not change the existing data value in the database. Essbase does not replace current values with empty values.

Valid Delimiters

You must separate fields from each other with delimiters. If you are loading data without a rules file, you must use spaces to delimit fields.

If you are using a rules file, delimiters can be any of the following:

- Tabs (default)
- Spaces
- New lines
- Carriage returns
- Commas

Extra Delimiters Without a Rules File

In data sources that are loaded without a rules file, Essbase ignores extra delimiters. In the following example, the fields are separated by spaces. Essbase ignores the extra spaces between the fields.

```
East    Cola    Actual    Jan    Sales    10
East    Cola    Actual    Feb    Sales    21
East    Cola    Actual    Mar    Sales    30
```

Extra Delimiters with a Rules File

In data sources that are loaded with a rules file, Essbase reads extra delimiters as empty fields. For example, if you try to use a rules file to load the file below into the Sample.Basic database, the load fails. Essbase reads the extra comma between East and Cola in the first record as an extra field. Essbase then puts Cola into Field 3. In the next record, however, Cola is in Field 2. Essbase expects Cola to be in Field 3 and stops the data load.

```
East,,Cola,Actual,Jan,Sales,10
East,Cola,Actual,Feb,Sales,21
East,Cola,Actual,Mar,Sales,30
```

To resolve the problem, delete the extra delimiter from the data source.

Valid Formatting Characters

Essbase views some characters in the data source as formatting characters only. Essbase ignores the characters listed in Table 38:

Table 38 Valid Formatting Characters

Formatting Character	Description
`==`	Multiple equal signs, such as for double underlining
`--`	Multiple minus signs, such as for single underlining
`_ _`	Multiple underscores
`==`	Multiple IBM PC graphic double underlines (ASCII character 205)
`_ _`	Multiple IBM PC graphic single underlines (ASCII character 196)

Ignored fields do not affect the data load or dimension build.

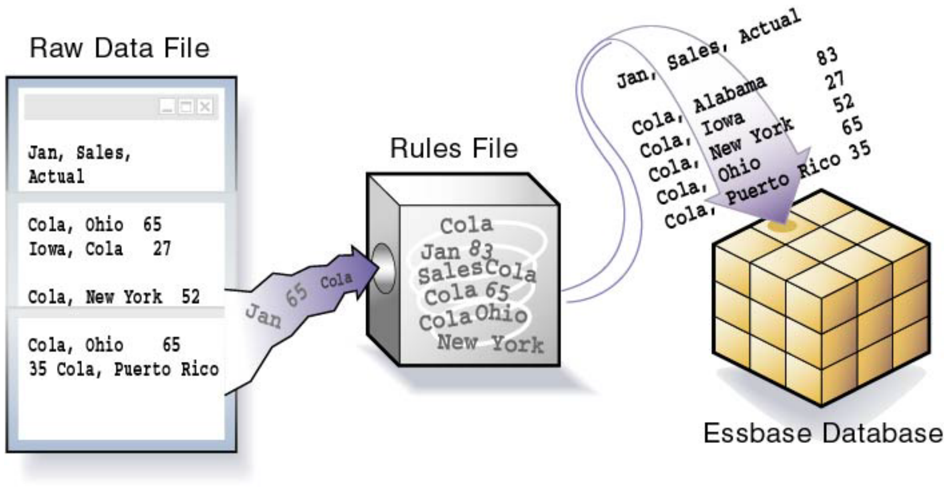
For example, Essbase ignores the equal signs in the following data source and loads the other fields normally.

```
East   Actual   "100-10"
       Sales    Marketing
       =====    =========
Jan    10       8
Feb    21       16
```

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.

Figure 62 Loading Data Sources Through Rules Files



Rules are stored in rules files. A rules file defines 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. It is best to create a separate rules file for each dimension.

Essbase reads the data values or members in the data source, changes them based on the rules in the rules file, and loads the changed data values into the database and the changed members into the outline. Essbase does not change the data source. You can reuse a rules file with any data source that requires the same set of rules.

After you create a dimension build rules file, you may want to automate the process of updating dimensions. See Appendix E, "Using ESSCMD."

Situations that Do and Do Not Need a Rules File

You need a rules file if the data source does not map perfectly to the database or if you are performing any of the following tasks:

- Loading data from a SQL data source
- Building dimensions
  - Adding dimensions and members to the database
  - Changing existing dimensions and members in the database
- Changing the data in any way, including the following:
  - Ignoring fields or strings in the data source
  - Changing the order of fields by moving, joining, splitting, or creating fields
  - Mapping the data in the data source to the database by changing strings
  - Changing the data values in the data source by scaling data values or by adding data values to existing data values in the data source
  - Setting header records for missing values
  - Rejecting an invalid record and continuing the data load

You do not need a rules file if you are performing a data load and the data source maps perfectly to the database. See "Data Sources that Do Not Need a Rules File" on page 264.

Note:

If you are using a rules file, each record in the rules file must have the same number of fields. See "Dealing with Missing Fields in a Data Source" on page 300.

Data Sources that Do Not Need a Rules File

If you are performing a dimension build, skip this section.

If a data source contains all of the information required to load the data values in it into the database, you can load the data source directly in a free-form data load.

To load a data value successfully, Essbase must encounter one member from each dimension before encountering the data value. For example, in Figure 61, Essbase loads the data value 42 into the database with the members Texas, 100-10, Jan, Sales, and Actual. If Essbase encounters a data value before a member of each dimension is specified, it stops loading the data source.

To map perfectly, a data source must contain all of the following and nothing else:

- One or more valid members from each dimension. A member name must be enclosed in quotation marks if it contains any of the following:
  - Spaces
  - Numeric characters (0–9)
  - Dashes (minus signs, hyphens)
  - Plus signs
  - Ampersands (&)

  If you are performing a data load without a rules file, when Essbase encounters an invalid member field, it stops the data load. Essbase loads all fields read before the invalid field into the database, resulting in a partial load of the data values. See "Loading Dimension Build and Data Load Error Logs" on page 755.
- One or more valid data values. See "Valid Data Fields" on page 261.

  If the data source contains blank fields for data values, replace the blank fields with #MI or #MISSING. Otherwise, the data values may not load correctly.
- Valid delimiters. See "Valid Delimiters" on page 262.

The fields in the data source must be formatted in an order that Essbase understands. The simplest way to format a record is to include a member from each dimension and a data field, as illustrated below:

```
Sales "100-10" Ohio Jan Actual 25
Sales "100-20" Ohio Jan Actual 25
Sales "100-30" Ohio Jan Actual 25
```

An incorrectly formatted data source will not load. You can edit the data source using a text editor and fix the problem. If you must perform many edits (such as moving several fields and records), consider using a rules file to load the data source. See "Rules Files" on page 263.

The following sections describe more complicated ways to format free-form data sources.

Formatting Ranges of Member Fields

If you are performing a dimension build, skip this section.

You can express member names as ranges within a dimension. For example, Sales and COGS form a range in the Measures dimension. Ranges of member names can handle a series of values.

A data source can contain ranges from more than one dimension at a time. In the example below, Jan and Feb form a range in the Year dimension and Sales and COGS form a range in the Measures dimension.

```
Actual Texas      Sales        COGS
```

```
                Jan      Feb      Jan     Feb
"100-10"        98       89       26      19
"100-20"        87       78       23      32
```

Notice that Sales is defined for the first two columns and COGS for the last two columns.

The following sections describe additional types of ranges.

Setting Ranges Automatically

If you are performing a dimension build, skip this section.

When Essbase encounters multiple members from the same dimension with no intervening data fields, it sets up a range for that dimension. The range stays in effect until Essbase encounters another member name from the same dimension, at which point Essbase replaces the range with the new member or new member range.

The following example contains a range of Jan to Feb in the Year dimension. It remains in effect until Essbase encounters another member name, such as Mar. When Essbase encounters Mar, the range changes to Jan, Feb, Mar.

```
Texas Sales
                    Jan    Feb    Mar
Actual    "100-10"  98     89     58
          "100-20"  87     78     115
```

Handling Out of Range Data Values

If you are performing a dimension build, skip this section.

When Essbase encounters a member range, it assumes that there is a corresponding range of data values. If the data values are not in the member range, the data load stops. Essbase loads any data fields read before the invalid field into the database, resulting in a partial data load.

The following example contains more data fields than member fields in the defined range of members. The data load stops when it reaches the 10 data field. Essbase loads the 100 and 120 data fields into the database.

```
Cola     Actual    East
           Jan     Feb
Sales      100     120     10
COGS       30      34      32
```

For information on restarting the load, see "Loading Dimension Build and Data Load Error Logs" on page 755.

Interpreting Duplicate Members in a Range

If you are performing a dimension build, skip this section.

Structure ranges in the source data so that Essbase interprets them correctly. If a member appears more than once in a range, Essbase ignores the duplicates.

The following example shows duplicate members for Actual, Budget, Sales, and Budget and two ranges: Actual to Budget and Sales to COGS. Essbase ignores the duplicate instances of Actual, Budget, Sales, and COGs (as shown in bold text).

```
Cola East
        Actual    Budget    Actual    Budget
        Sales     Sales     COGS      COGS
Jan     108       110       49        50
Feb     102       120       57        60
```

For Actual, the first member of the first range, Essbase maps data values to each member of the second range (Sales and COGS). Essbase then proceeds to the next value of the first range, Budget, similarly mapping values to each member of the second range. As a result, Essbase interprets the file as shown below:

```
Cola East
           Actual              Budget
           Sales      COGS     Sales     COGS
Jan        108        110      49        50
Feb        102        120      57        60
```

Reading Multiple Ranges

If you are performing a dimension build, skip this section.

As Essbase scans a file, it processes the most recently encountered range first when identifying a range of data values. The example above contains two ranges: Actual and Budget and Sales and COGS. While reading the file from left to right and top to bottom, Essbase encounters the Actual and Budget range first and the Sales and COGS range second. Because the Sales and COGS range is encountered second, Essbase puts data fields in the Sales and COGS part of the database first.

Formatting Columns

If you are performing a dimension build, skip this section.

Files can contain columns of fields. Essbase supports loading data from symmetric columns or asymmetric columns.

Symmetric Columns

If you are performing a dimension build, skip this section. Dimension builds require a rules file.

Symmetric columns have the same number of members under them. In the following example, each dimension column has one column of members under it. For example, Product has one column under it (100-10 and 100-10) and Market has one column under it (Texas and Ohio).

```
Product    Measures    Market    Year    Scenario
"100-10"   Sales       Texas     Jan     Actual      112
"100-10"   Sales       Ohio      Jan     Actual      145
```

The columns in the following file are also symmetric, because Jan and Feb have the same number of members under them:

```
                                  Jan             Feb
                            Actual   Budget   Actual   Budget
"100-10"   Sales   Texas   112      110      243      215
"100-10"   Sales   Ohio    145      120      81       102
```

Asymmetric Columns

If you are performing a dimension build, skip this section.

Asymmetric columns have different numbers of members under them. In the following example, the Jan and Feb columns are asymmetric because Jan has two columns under it (Actual and Budget) and Feb has one column under it (Budget):

```
                             Jan      Jan      Feb
                             Actual   Budget   Budget
"100-10"   Sales   Texas    112      110      243
"100-10"   Sales   Ohio     145      120      81
```

If a file contains asymmetric columns, label each column with the appropriate member name.

The example above is valid because the Jan label is now over Actual and Budget. It is clear to Essbase that both columns map to Jan.

The following example is not valid because the column labels are incomplete. The Jan label must appear over the Actual and Budget columns.

```
                             Jan               Feb
                             Actual   Budget   Budget
"100-10"   Sales   Texas    112      110      243
"100-10"   Sales   Ohio     145      120      81
```

Security and Multiple-User Considerations

Essbase supports concurrent multiple users reading and updating the database; therefore, users can use the database while you are dynamically building dimensions, loading data, or calculating the database. In a multi-user environment, Essbase protects data by using the security system described in Chapter 38, “User Management and Security”.

- Security Issues

  The security system prevents unauthorized users from changing the database. Only users with write access to a database can load data values or add dimensions and members to the database. Write access can be provided globally or by using filters.

- Multi-User Data Load Issues

  You can load data values while multiple users are connected to a database. Essbase uses a block locking scheme for handling multi-user issues. When you load data values, Essbase does the following:

  - Locks the block it is loading into so that no one can write to the block.

    See Chapter 48, “Ensuring Data Integrity” for information on Essbase transaction settings, such as identifying whether other users get read-only access to the locked block or noting how long Essbase waits for a locked block to be released.

  - ❍ Updates the block.

    See “Data Locks” on page 780 for information on whether Essbase unlocks a block when its update is complete or waits for the entire data load to complete before unlocking the block.

- Multi-User Dimension Build Issues

  You cannot build dimensions while other users are reading or writing to the database. After you build dimensions, Essbase restructures the outline and locks the database for the duration of the restructure operation.

17 Working with Rules Files

In This Chapter

Also see:

- Chapter 16, "Understanding Data Loading and Dimension Building"
- Chapter 19, "Performing and Debugging Data Loads or Dimension Builds"

Process for Creating Data Load Rules Files

To create a data load rules file:

1. Determine whether to use the same rules file for data loads and dimension builds.

   See "Combining Data Load and Dimension Build Rules Files" on page 272.

2. Create a rules file.

   See "Creating Rules Files" on page 273.

3. Set the file delimiters for the data source.

   See "Setting File Delimiters" on page 274.

4. Map each rules file field to the data source and define field properties.

   See "Defining Data Load Field Properties" on page 280.

5. If necessary, set record, field, and data operations to change the data in the data source during loading.

   See Chapter 18, "Using a Rules File to Perform Operations on Records, Fields, and Data."
6. Validate and save the rules file.

   See "Setting Dimension Build Operational Instructions" on page 280.

Process for Creating Dimension Build Rules Files

To create a dimension build rules file:

1. Determine whether to use the same rules file for data loads and dimension builds.

   See "Combining Data Load and Dimension Build Rules Files" on page 272.
2. Create a rules file.

   See "Creating Rules Files" on page 273.
3. Set the file delimiters for the data source.

   See "Setting File Delimiters" on page 274.
4. If you are creating a dimension, name the dimension.

   See "Naming New Dimensions" on page 274.
5. Select the build method.

   See "Selecting a Build Method" on page 274.
6. If necessary, change or set the properties of members and dimensions you are building.

   See "Setting and Changing Member and Dimension Properties" on page 275.
7. If necessary, set record and field operations to change the members in the data source during loading.

   See Chapter 18, "Using a Rules File to Perform Operations on Records, Fields, and Data."
8. Set field type information, including field type, field number, and dimension.

   See "Setting Field Type Information" on page 277.
9. Validate and save the rules file.

   See "Setting Dimension Build Operational Instructions" on page 280.

Combining Data Load and Dimension Build Rules Files

Before building a rules file, you must decide whether the rules file will be used for data loads and dimension builds, for only data loads, or for only dimension builds. Once you create a rules file, you cannot separate it into two rules files. Similarly, you cannot merge two rules files into one file.

Use the same rules file for both data load and dimension build if you plan to load the data source and build new dimensions simultaneously.

Use separate rules files for data load and dimension build under any of the following circumstances:

- To build an outline from scratch
- To perform different field operations during the data load and dimension build
- To reuse the data load or dimension build rules file separately
- To use data sources that contain no data values, only dimensions

Creating Rules Files

A rules files tells Essbase what changes to make to the data source and outline during a data load or dimension build.

Note:

In rules files, record size is limited to 64 KB.

➤ To create a rules file:

1 **If you are creating the rules file on the Essbase Server, connect to the server.**

Connecting to the server is not necessary if you are creating the rules file on the client.

2 **Open Data Prep Editor.**

See “Creating a Rules File” or “Opening an Existing Rules File” in the *Oracle Essbase Administration Services Online Help*.

You can open Data Prep Editor with a new or existing rules file. After you open Data Prep Editor, put the editor in the correct mode. See “About Data Prep Editor” in the *Oracle Essbase Administration Services Online Help*.

3 **Open the data source.**

In Data Prep Editor, you can open data sources such as text files, spreadsheet files, and SQL data sources. Data Prep Editor displays the data source, enabling you to see what needs to be changed.

- To open text files and spreadsheet files, see “Opening a Data File” in the *Oracle Essbase Administration Services Online Help*.
- To open SQL data sources, see “Opening an SQL Database” in the *Oracle Essbase Administration Services Online Help*.

  To open an SQL data source, use SQL Interface. The *Oracle Essbase SQL Interface Guide* provides information on supported environments, installation, and connection to supported data sources. Contact your Essbase administrator for more information.

  You can define a substitution variable for the data source name (DSN). When you open a SQL data source, you can select the substitution variable for the value you want to use as the DSN. For example, you can create a substitution variable named Payroll_detail and create a rules file that specifies Payroll_detail as the substitution variable for the data source name. Before performing the data load or dimension build, you must set the value for Payroll_detail to the data source name you want to use; for example, an Oracle or IBM DB2 database. When a data load or dimension build is performed, the substitution variable value that Essbase Server finds at that time is used. See "Using Substitution Variables" on page 120.

Note:

When you open an SQL data source, the rules fields default to the SQL data source column names. If the names are not the same as the Essbase dimension names, map the fields to the dimensions. See "Changing Field Names" on page 291.

Setting File Delimiters

A file delimiter is the character (or characters) used to separate fields in the data source. By default, a rules file expects fields to be separated by tabs. You can set the file delimiter as a comma, tab, space, fixed-width column, or custom value. Acceptable custom values are characters in the standard ASCII character set, numbered from 0 through 127. Usually, setting file delimiters is what you do first after opening a data source.

Note:

You need not set file delimiters for SQL data.

➤ To set file delimiters, see "Setting File Delimiters" in the *Oracle Essbase Administration Services Online Help*.

Naming New Dimensions

If you are not creating a dimension in the rules file, skip this section.

If you are creating a dimension, you must name it in the rules file. See "Naming Restrictions for Dimensions, Members, and Aliases" on page 1060.

If you are creating an attribute dimension, the base dimension must be a sparse dimension already defined in the outline or the rules file. See Chapter 10, "Working with Attributes."

➤ To name a new dimension, see "Creating a Dimension Using a Rules File" in the *Oracle Essbase Administration Services Online Help*.

Selecting a Build Method

If you are not performing a dimension build, skip this section.

If you are building a new dimension or adding members to an existing dimension, you must specify a build method for each dimension that you are creating or modifying. For information about each build method, see "Understanding Build Methods" on page 309.

➤ To select a build method, see "Choosing a Build Method" in the *Oracle Essbase Administration Services Online Help*.

Setting and Changing Member and Dimension Properties

If you are not performing a dimension build, skip this section.

If you are performing a dimension build, you can set or change the properties of the members and dimensions in the outline. Some changes affect all members of the selected dimension, some affect only the selected dimension, and some affect all dimensions in the rules file.

You can set or change member and dimension properties using the Data Prep Editor or change the member properties in the data source.

Using Data Prep Editor to Set Dimension and Member Properties

If you are not performing a dimension build, skip this section.

➤ To set dimension properties, see "Setting Dimension Properties" in the *Oracle Essbase Administration Services Online Help*.

➤ To set member properties, see "Setting Member Properties" in the *Oracle Essbase Administration Services Online Help*.

Using the Data Source to Work with Member Properties

You can modify the properties of new and existing members during a dimension build by:

- Including member properties in a field in the data source
- Leaving the data source field empty to reset the property to the default value, or to remove the formula or UDA

In Administration Services Console, the following dimension build options control whether the value in the data source property field is applied to the associated member:

- Allow property changes
- Allow formula changes
- Allow UDA changes

In the data source, put the properties in the field directly following the field containing the members that the properties modify. For example, to specify that the `Margin%` member not roll up into its parent and not be shared:

1. Position the ~ property (which indicates that the member should not roll up into its parent) and the N property (which indicates that the member should not be shared) after the Margin % field. For example:

   ```
   Margin% Margin% ~ N Sales
   ```

2. Set the field type for the properties fields to Property.

   See "Setting Field Type Information" on page 277.

Removing a formula, UDA, or attribute, or resetting a property to its default value, includes the following additional steps:

- In Administration Services Console, select the Delete when the field is empty option for the Property field on the Dimension Build Properties tab of the Field Properties dialog box. (This option is ignored if the appropriate dimension property is not selected in the Dimension Build dialog box.)
- Leave the field NULL or empty in the data source.

Table 39 lists all member codes used in the data source to assign properties to block storage outline members. (For a list of properties that can be assigned to aggregate storage outline members, see "Rules File Differences for Aggregate Storage Dimension Builds" on page 955.)

Table 39 Member Property Codes

Code	Description
%	Express as a percentage of the current total in a consolidation
*	Multiply by the current total in a consolidation
+	Add to the current total in a consolidation
-	Subtract from the current total in a consolidation
/	Divide by the current total in a consolidation
~	Exclude from the consolidation
^	Exclude from all consolidations in all dimensions
A	Treat as an average time balance item (applies to accounts dimensions only)
B	Exclude data values of zero or #MISSING in the time balance (applies to accounts dimensions only)
E	Treat as an expense item (applies to accounts dimensions only)
F	Treat as a first time balance item (applies to accounts dimensions only)
L	Treat as a last time balance item (applies to accounts dimensions only)
M	Exclude data values of #MISSING from the time balance (applies to accounts dimensions only)

Code	Description
N	Never allow data sharing
O	Tag as label only (store no data)
S	Set member as stored member (non-Dynamic Calc and not label only)
T	Require a two-pass calculation (applies to accounts dimensions only)
V	Create as Dynamic Calc and Store
X	Create as Dynamic Calc
Z	Exclude data values of zero from the time balance (applies to accounts dimensions only)

Setting Field Type Information

If you are not performing a dimension build, skip this section.

In a dimension build, each field in the data source is part of a column that describes an outline member. Fields can contain information about:

- Member names
- Member properties
- Attribute associations

For Essbase to process this information, you must specify the following information when setting field types:

- Field type

  The type of field to expect in that column, such as a generation field or an alias field. The field type depends on the data source and the build method (see "Understanding Build Methods" on page 309).

- Dimension

  The dimension to which the members of that column belong.

- Generation or level number

  The generation or level number of the members of that column.

➤ To set field type information, see "Setting Field Types" in the *Oracle Essbase Administration Services Online Help*.

Field Types and Valid Build Methods

Table 40 lists field types and valid build methods.

Table 40 Field Types and Valid Build Methods

Field Type[*]	What the Field Contains	Valid Build Methods
Alias	An alias **Note:** The alias value will not be assigned to the new member if Member update dimension build is set to Remove unspecified and the data source for a new member contains the alias value of a removed member.	Generation, level, and parent-child references See "Rules for Assigning Field Types" on page 279.
Property	A member property. See Table 39, "Member Property Codes," on page 276.	
Formula	A formula	
Currency name	(Block storage outlines only) A currency name	
Currency category	(Block storage outlines only) A currency category	
UDA	A UDA	
Attribute parent	In an attribute dimension, the name of the parent member of the attribute member in the following field	
The name of a specific attribute dimension	A member of the specified attribute dimension. This member is associated with a specified generation or level of the selected base dimension.	
Generation	The name of a member in the specified generation	Generation references
Duplicate generation	The name of a member with a shared member as a child	
Duplicate generation alias	The alias for the shared member	
Level	The name of a member in a level	Level references
Duplicate level	The name of a member with a shared member as a child	
Duplicate level alias	The alias for the shared member	
Parent	The name of a parent	Parent-child reference
Child	The name of a child	

[*]Field types whose names begin with duplicate (such as duplicate generation and duplicate level alias), are not related to duplicate member names described in Chapter 8, "Creating and Working With Duplicate Member Outlines".

Rules for Assigning Field Types

Table 41 lists the rules for selecting valid field types, depending on the build method.

Table 41 Rules for Assigning Field Types Based on Build Method

Build Method	Rules for Assigning Field Types

Generation	● If GEN numbers do not start at 2, the first member of the specified generation must exist in the outline.
	● GEN numbers must form a contiguous range. For example, if GEN 3 and GEN 5 exist, you must also define GEN 4.
	● Put DUPGEN fields immediately after GEN fields.
	● Put DUPGENALIAS fields immediately after DUPGEN fields.
	● Group GEN fields sequentially within a dimension. For example:
	`GEN2,PRODUCT    GEN3,PRODUCT    GEN4,PRODUCT`
	● Put attribute association fields after the base field with which they are associated, and specify the generation number of the associated base dimension member. For example:
	`GEN2,PRODUCT    GEN3,PRODUCT    OUNCES3,PRODUCT`
	The generation number must correspond to the generation of the member in the outline for which the field provides values. For example, the `3` in `GEN3,PRODUCT` shows that the values in the field are third-generation members of the Product dimension. The `2` in `ALIAS2,POPULATION` shows that the values in the field are associated with the second-generation member of the Population dimension.

Level	● Put DUPLEVEL fields immediately after LEVEL fields. ● Put DUPLEVELALIAS fields immediately after the DUPLEVEL fields. ● Each record must contain a level 0 member. If a level 0 member is repeated on a new record with a different parent, Essbase rejects the record unless you select the Allow Moves member property. See "Setting Member Properties" in the *Oracle Essbase Administration Services Online Help*. ● Group level fields sequentially within a dimension. ● Put the fields for each roll-up in sequential order. ● Use a single record to describe the primary and secondary roll-ups. ● Put attribute association fields after the base field with which they are associated, and specify the level number of the associated base dimension member. For example: `LEVEL3,PRODUCT    OUNCES3,PRODUCT    LEVEL2,PRODUCT` ● The level number must correspond to the level of the member in the outline for which the field provides values. For example, the 3 in LEVEL3,PRODUCT shows that the values in the field are level 3 members of the Product dimension. The `2` in `ALIAS2,POPULATION` shows that the values in the field are associated with the second level of the Population dimension.
Parent-child	If field type is parent or child, enter 0 (zero) in the Number text box.
Attribute dimension name	The generation or level number must correspond to the generation or level of the associated base member in the outline. For example, the `3` in `OUNCES3,PRODUCT` shows that the values in the field are the members of the Ounces attribute dimension that are associated with the third-generation member of the Product dimension in the same source data record.

If necessary, move the fields to the required locations. See "Moving Fields" on page 289.

➤ To move fields, see “Moving Fields” in the *Oracle Essbase Administration Services Online Help*.

Setting Dimension Build Operational Instructions

If you are not performing a dimension build, skip this section.

Within the rules file, you define operations to be performed after the data source has been read:

- Whether to sort members after Essbase has processed and added all members from the data source
- Whether to add the members to the existing outline or to remove unspecified members from the outline

  Removing unspecified members is available only with the generation reference, level reference, and parent-child reference build methods.

Note:

Outlines are invalid if removing members results in level 0 Dynamic Calc members without formulas.

Defining Data Load Field Properties

You must map each rules file field to the corresponding outline member, or as a data field or ignored field. Other field characteristics may also apply.

For duplicate member outlines, you must specify the method (level reference, generation reference, or dimension reference) that Essbase uses to map the field.

- Level and generation references: The data source contains multiple fields within the duplicate member dimension to uniquely identify duplicate members.
  - Use the level reference method when fields within a dimension are organized bottom-up in the data source.
  - Use the generation reference method when fields within a dimension are organized top-down in the data source. For example:

```
gen2,Market, gen3,Market, Product, Year, Measures, Scenario, *data*
State,"New York","100-10",Jan,Sales,Actual,42
City,"New York","100-20",Jan,Sales Actual,82
State,Texas,"100-10",Jan,Sales,Actual,37
```

- Dimension reference: If an outline contains a duplicate member name in different dimensions—for example, a member name such as Other can be meaningful in different dimensions—you can use the dimension reference method. When you set a field to use the dimension reference method, you also identify the dimension to which members in that field belong. When the dimension reference method is specified for a field, Essbase checks to ensure that members in the field belong to the dimension specified for the field.

➤ To specify generation, level, or dimension references for data loads, see “Mapping Field Names” in the *Oracle Essbase Administration Services Online Help*.

Performing Operations on Records, Fields, and Data

A rules file enables you to perform operations on records, fields, and data values before loading them into the database without changing the data source. See Chapter 18, “Using a Rules File to Perform Operations on Records, Fields, and Data.”

Validating, Saving, and Printing

Rules files are validated to ensure that the members and dimensions in the rules file map to the outline. Validation cannot ensure that the data source loads properly.

➤ To validate a rules file, see “Validating a Rules File” in the *Oracle Essbase Administration Services Online Help*.

If the rules file is correct, you can perform a data load or dimension build. See Chapter 19, “Performing and Debugging Data Loads or Dimension Builds.”

If the rules file is not valid, see the appropriate topic for each rules file type:

- Data load: “Requirements for Valid Data Load Rules Files” on page 281
- Dimension build: “Requirements for Valid Dimension Build Rules Files” on page 282

➤ To save a rules file, see “Saving a Rules File” in the *Oracle Essbase Administration Services Online Help*.

Requirements for Valid Data Load Rules Files

For a data load rules file to validate, all of the following questions must be answered “yes.”

- Is the rules file associated with the correct outline?

  See “Validating a Rules File” in the *Oracle Essbase Administration Services Online Help*.
- Does each record in the data source contain only one member from each dimension?

  See “Items in a Data Source” on page 259.
- Are all member and dimension names spelled correctly?
- Are all members surrounded by quotation marks if they contain numbers or file delimiters?

  See “Valid Member Fields” on page 260.
- Are there no extra delimiters in the data source?

  See “Extra Delimiters with a Rules File” on page 262.
- Is the member that each data field maps to spelled correctly in the rules file?

See “Changing Field Names” on page 291.

- Are the file delimiters correctly placed?

  See “Valid Delimiters” on page 262.
- Is the member in the field name a valid member?

  See “Mapping Fields” on page 291.
- Is the dimension name used in only one field (for example, not in a field name and the header)?

  You can map a single data value to only one set of members.
- Is only one field defined as a data field?

  See “Defining a Column as a Data Field” on page 293.
- Is the UDA used for sign flipping in the associated outline?

  See “Flipping Field Signs” on page 295.

Requirements for Valid Dimension Build Rules Files

For a dimension build rules file to validate, all of the following questions must be answered “yes.”

- Is the rules file associated with the correct outline?

  See “Validating a Rules File” in the *Oracle Essbase Administration Services Online Help*.
- Does each record contain only one member from each dimension?

  See “Items in a Data Source” on page 259.
- Are all member and dimension names spelled correctly?
- Are all members enclosed in quotation marks if they contain numbers or file delimiters?

  See “Valid Member Fields” on page 260.
- Are there no extra delimiters in the data source?

  See “Extra Delimiters with a Rules File” on page 262.
- Are the reference numbers sequential?

  See “Rules for Assigning Field Types” on page 279.
- Are there no repeated generations?

  See “Rules for Assigning Field Types” on page 279.
- Is each field type valid for the build method?

  See “Field Types and Valid Build Methods” on page 277.
- Are all the fields in correct order?

  See “Rules for Assigning Field Types” on page 279.
- Does each child field have a parent field?
- Do all dimension names exist in the outline or the rules file?

- Are any dimensions specified in both the header record in the rules file and the header record in the data source?

  Dimensions can be specified in either the header in the rules file or the header in the data source, but not in both. See "Defining Header Records" on page 286.

Copying Rules Files

You can copy rules files to applications and databases on any Essbase Server, according to your permissions. You can also copy rules files across servers as part of application migration.

➤ To copy a rules file, use a tool:

Tool	Topic	Location
Administration Services	Copying a Rules File	*Oracle Essbase Administration Services Online Help*
MaxL	**alter object**	*Oracle Essbase Technical Reference*
ESSCMD	COPYOBJECT	*Oracle Essbase Technical Reference*

Printing Rules Files

You can print the entire contents and properties of a data load or dimension build rules file. You can also specify properties and settings to print.

➤ To print a rules file, see "Printing Rules Files" in the *Oracle Essbase Administration Services Online Help*.

18 Using a Rules File to Perform Operations on Records, Fields, and Data

In This Chapter

Also see:

- Chapter 16, "Understanding Data Loading and Dimension Building"
- Chapter 17, "Working with Rules Files"

Performing Operations on Records

You can perform operations at the record level. For example, you can reject certain records before they are loaded into the database. See the following sections.

Selecting Records

You can specify which records Essbase loads into the database or uses to build dimensions by setting selection criteria. *Selection criteria* are string and number conditions that must be met by one or more fields within a record for Essbase to load the record. If a field or fields in the record do not meet the selection criteria, Essbase does not load the record. You can define one or more selection criteria. For example, to load only 2003 Budget data from a data source, create a selection criterion to load only records in which the first field is Budget and the second field is 2003. If you define selection criteria on more than one field, you can specify how Essbase combines the criteria. See "Combining Multiple Select and Reject Criteria" on page 286.

➤ To select a record, see "Selecting Records" in the *Oracle Essbase Administration Services Online Help*.

Rejecting Records

You can specify which records Essbase ignores by setting rejection criteria. *Rejection criteria* are string and number conditions that, when met by one or more fields within a record, cause Essbase to reject the record. You can define one or more rejection criteria. If no field in the record meets the rejection criteria, Essbase loads the record. For example, to reject Actual data from a data source and load only Budget data, create a rejection criterion to reject records in which the first field is Actual.

➤ To reject a record, see “Rejecting Records” in the *Oracle Essbase Administration Services Online Help*.

Combining Multiple Select and Reject Criteria

When you define select and reject criteria on multiple fields, you can specify how Essbase combines the rules across fields: whether the criteria are connected logically with AND or with OR. If you select AND from the Boolean group, the fields must match all of the criteria. If you select OR, the fields must match only one of the criteria. The global Boolean setting applies to all select or reject operations in the rules file, for data load and dimension build fields.

Note:

If selection and rejection criteria apply to the same record (you define select and reject criteria on the same record), the record is rejected.

➤ To determine how to combine select and reject criteria on multiple fields, see “Combining Selection and Rejection Criteria” in the *Oracle Essbase Administration Services Online Help*.

Setting the Records Displayed

You can specify the number of records, and the first record, that Essbase displays in Data Prep Editor. When you specify the first record, Essbase skips all preceding records. For example, if you enter 5 as the starting record, Essbase does not display records 1 through 4.

Note:

Essbase treats header records the same as data records when counting the records to skip.

➤ To set the records displayed, see “Setting the Records Displayed” in the *Oracle Essbase Administration Services Online Help*.

Defining Header Records

Data sources can contain:

- *Data records*, which contain member fields and data fields
- *Header records*, which describe the contents of the data source and how to load values from the data source to the database

Rules files contain records that translate the data of the data source to map it to the database. As part of that information, rules files can also contain header records. For example, the Sample.Basic database has a dimension for Year. If several data sources arrive with monthly numbers from different regions, the month itself might not be specified in the data sources. You must set header information to specify the month.

You can create a header record using one of the following methods:

- Define header information in the rules file.

  Rules file headers are used only during data loading or dimension building and do not change the data source. Header information set in a rules file is not used if the rules file also points to header records in the data source.

- Define header information in the data source and, in the rules file, point to the header records.

  Placing header information in the data source makes it possible to use the same rules file for multiple data sources with different formats, because the data source format is specified in the data source header (not in the rules file).

  When you add one or more headers to the data source, you must also specify in the rules files the location of the headers in the data source. The rules file tells Essbase to read the header information as a header record (not as a data record). You can also specify the type of header information in each header record.

  Header information defined in the data source takes precedence over header information defined in the rules file.

➤ To define a header in the rules file, see "Setting Headers in the Rules File" in the *Oracle Essbase Administration Services Online Help*.

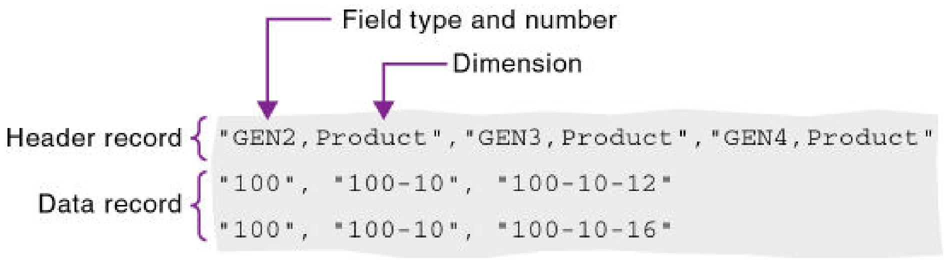
➤ To define a header in the data source, see "Setting Headers in the Data Source" in the *Oracle Essbase Administration Services Online Help*.

Data Source Headers

You can build dimensions dynamically by adding header information to the top record of the data source and by specifying the location of the header record in the rules file.

The header record lists field definitions for each field. The field definition includes the field type, the field number, and the dimension name into which to load the fields. The format of a header record is illustrated below:

Figure 63   Header Record with Three Field Definitions



```
"GEN2,Product","GEN3,Product","GEN4,Product"
"100",  "100-10",  "100-10-12"
"100",  "100-10",  "100-10-16"
```

If the file delimiter is a comma, enclose each field definition in quotation marks (" ").

After you set the header information in the data source, you must specify the location of the header information in the rules file. If a rules file refers to header information in a data source, Essbase uses the information in the data source—rather than the information in the rules file—to determine field types and dimensions.

Valid Data Source Header Field Types

Valid field types, which must be in capital letters:

- GEN, DUPGEN, and DUPGENALIAS
- LEVEL, DUPLEVEL, and DUPLEVELALIAS
- PARENT, CHILD
- PROPERTY
- ALIAS
- FORMULA
- CURNAME
- CURCAT
- UDA
- ATTRPARENT
- The name of an attribute dimension, such as CAFFEINATED

Each field type that you set requires a field number. When the field type is the name of an attribute dimension, the field number cannot be greater than 9. See "Setting Field Type Information" on page 277.

Performing Operations on Fields

You can perform operations at the field level. For example, you can move a field to a new position in the record. See the following sections.

Ignoring Fields

You can ignore all fields of a specified column of the data source. The fields still exist in the data source, but they are not loaded into the Essbase database. For example, the Sample.Basic database has five standard dimensions: Year, Product, Market, Measures, and Scenario. If the data source has an extra field that is not a member of any dimension, such as Salesperson, you can tell Essbase to ignore the Salesperson field.

➤ To ignore all fields in a column, see "Ignoring Fields" in the *Oracle Essbase Administration Services Online Help*.

Ignoring Strings

You can ignore any field in the data source that matches a string, called a *token*. When you ignore fields based on string values, the fields are ignored everywhere they appear in the data source, not just in a particular column. For example, in a data source that is a computer-generated report in text format, special ASCII characters might be used to create horizontal lines between pages or boxes around headings. These special characters can be defined as tokens to be ignored.

➤ To ignore all instances of a string, see "Ignoring Fields Based on String Matches" in the *Oracle Essbase Administration Services Online Help*.

Arranging Fields

You can set the order of the fields in the rules file to be different from the order of the fields in the data source. The data source is unchanged. See the following sections.

Moving Fields

You can move fields to a different location using a rules file. For example, you can specify the first field in the data source to be the third field during the data load or dimension build.

In some instances, moved fields may appear to merge. If you move a field that contains empty cells, and the moved field becomes the last field in the record, as shown below, the field may merge with the field to its left.

```
1<tab>2<tab>3
1<tab>2<tab>(null)
```

To prevent merging, replace the empty cell with a delimiter.

➤ To move fields, see “Moving Fields” in the *Oracle Essbase Administration Services Online Help*.

Joining Fields

You can join multiple fields into one field. The new field is given the name of the first field in the join. For example, if a data source has separate fields for product number (100) and product family (-10), you must join the fields (100-10) before loading them into the Sample.Basic database.

Before you join fields, move the fields to join into the order in which you want them joined. See “Moving Fields” in the *Oracle Essbase Administration Services Online Help*.

➤ To join fields, see “Joining Fields” in the *Oracle Essbase Administration Services Online Help*.

Creating a Field by Joining Fields

You can join fields by placing the joined fields into a new field. This procedure leaves the original fields intact. Creating a field is useful if you need to concatenate fields of the data source to create a member.

For example, if a data source has separate fields for product number (100) and product family (-10), you must join the fields (100-10) before you load them into the Sample.Basic database. If, however, you want to preserve the two existing fields in the data source, you can create a field (100-10) using a join. The data source now includes all three fields (100, -10, and 100-10).

Before you join fields, move the fields to join into the order in which you want them joined. See “Moving Fields” in the *Oracle Essbase Administration Services Online Help*.

➤ To create a field by joining existing fields, see “Creating a Field Using Joins” in the *Oracle Essbase Administration Services Online Help*.

Copying Fields

You can create a copy of a field while leaving the original field intact. For example, if, during a single dimension build, you want to define a multilevel attribute dimension and associate attributes with members of a base dimension, you must copy some of the fields. See “Working with Multilevel Attribute Dimensions” on page 322.

➤ To copy a field, select one field and then create a field using a join; see “Creating a Field Using Joins” in the *Oracle Essbase Administration Services Online Help*.

Splitting Fields

You can split a field into two fields. For example, if a data source for the Sample.Basic database has a field containing UPC100-10-1, you can split “UPC” out of the field and ignore it. Then, only 100-10-1, the product number, is loaded.

➤ To split a field, see “Splitting Fields” in the *Oracle Essbase Administration Services Online Help*.

Creating Additional Text Fields

You can create a text field between two fields. You might create a text field to insert text between fields that are to be joined. For example, if one field contains 100 and one contains 10-1, you can insert a text field with a dash between the two fields and then join the three fields to create the 100-10-1 member of the Product dimension.

➤ To create a field and populate it with text, see “Creating a Field Using Text” in the *Oracle Essbase Administration Services Online Help*.

Undoing Field Operations

You can undo the last field operation that you performed, such as move, split, join, create using text, or create using join, by using Undo command (select Edit, then Undo). You can undo field operations even if you have performed other actions. Undoing field operations is sequential, starting with the most recently performed operation.

➤ To undo one or more field operations, see “Undoing Field Operations” in the *Oracle Essbase Administration Services Online Help*.

Mapping Fields

This section applies to data load only. If you are performing a dimension build, skip this section.

You use a rules file to map data source fields to Essbase member names during a data load. You can map fields in a data source directly to fields in the Essbase database during a data load by specifying which field in the data source maps to which member or member combination in the Essbase database. The data source is not changed.

Note:

When you open a SQL data source, the fields default to the SQL data source column names. If the SQL column names and the Essbase dimension names are the same, you need not map the column names.

➤ To map fields, see “Mapping Field Names” in the *Oracle Essbase Administration Services Online Help*.

Changing Field Names

To load a data source, you must specify how the fields of the data source map to the dimensions and members of the database. Rules files can translate fields of the data source so that the fields match member names each time the data source is loaded. This process does not change the data source.

The rules file:

- Maps member fields of the data source to dimensions and members of the database
- Maps data fields of the data source to member names or member combinations (such as Jan, Actual) of the database

See the following sections.

Replacing Text Strings

You can replace text strings so that the fields map to Essbase member names during a data load or dimension build. The data source is not changed. For example, if the data source abbreviates New York to NY, you can have the rules file replace each NY with New York.

➤ To replace a text string, see “Replacing Field Names” in the *Oracle Essbase Administration Services Online Help*.

Replacing an Empty Field with Text

You may want to replace empty fields in a column with text. For example, if empty fields in the column represent default values, you can insert the default values or insert `#MI` to represent missing values.

➤ To replace an empty field with text, see “Replacing an Empty Field with Text” in the *Oracle Essbase Administration Services Online Help*.

Changing the Case of Fields

You can change the case of a field so that the field maps to Essbase member names during a data load or dimension build. The data source is not changed. For example, if the data source capitalizes a field (for example, JAN) that is in lowercase in the database (jan), you can have the rules file change the field to lowercase.

➤ To change the case of values in a field, see “Changing Case of Fields” in the *Oracle Essbase Administration Services Online Help*.

Dropping Leading and Trailing Spaces

You can drop leading and trailing spaces from around fields of the data source. A field value containing leading or trailing spaces does not map to a member name, even if the name within the spaces is an exact match.

By default, Essbase drops leading and trailing spaces.

➤ To drop spaces around a field, see "Dropping Spaces Around Fields" in the *Oracle Essbase Administration Services Online Help*.

Converting Spaces to Underscores

You can convert spaces in fields of the data source to underscores to make the field values match the member names of the database.

➤ To change spaces to underscores, see "Converting Spaces to Underscores" in the *Oracle Essbase Administration Services Online Help*.

Adding Prefixes or Suffixes to Field Values

You can add prefixes and suffixes to each field value of the data source. For example, you can add 2002 as the prefix to all member names in the Year dimension.

➤ To add prefix or suffix values to a field, see "Adding Prefixes and Suffixes" in the *Oracle Essbase Administration Services Online Help*.

Performing Operations on Data

This section applies to data load only. If you are performing a dimension build, skip this section.

You can perform operations on the data in a field; for example, moving a field to a new position in the record. See the following sections.

Defining a Column as a Data Field

This section applies to data load only. If you are performing a dimension build, skip this section.

If each record in the data source contains a column for every dimension and one data column, you must define the data column as a data field, as shown in the following example:

```
Market, Product, Year, Measures, Scenario
Texas  100-10  Jan  Sales Actual 42
Texas  100-20  Jan  Sales Actual 82
Texas  100-10  Jan  Sales Actual 37
```

You can define only one field in a record as a data field.

➤ To define a data field, see "Defining a Column as a Data Field" in the *Oracle Essbase Administration Services Online Help*.

Adding to and Subtracting from Existing Values

This section is for data load only. If you are performing a dimension build, skip this section.

By default, Essbase overwrites the existing values of the database with the values of the data source, but you can determine how newly loaded data values affect existing data values.

You can use incoming data values to add to or subtract from existing database values. For example, if you load weekly values, you can add them to create monthly values in the database.

Using this option makes recovery more difficult if the database crashes while loading data, although Essbase lists the number of the last row committed in the application log. See "Contents of the Application Log" on page 733.

To prevent difficult recoveries, set the Commit Row database transaction option to 0. This setting causes Essbase to view the entire load as a single transaction and to commit the data only when the load is complete. See "Understanding Isolation Levels" on page 779.

➤ To add to existing data values, see "Adding to Data Values" in the *Oracle Essbase Administration Services Online Help*.

➤ To subtract from existing data values, see "Subtracting from Data Values" in the *Oracle Essbase Administration Services Online Help*.

Clearing Existing Data Values

This section is for data load only. If you are performing a dimension build, skip this section.

You can clear existing data values from the database before you load new values. By default, Essbase overwrites the existing values of the database with the new values of the data source. If you are adding and subtracting data values, however, Essbase adds or subtracts the new data values to and from the existing values.

Before adding or subtracting new values, make sure that the existing values are correct. Before loading the first set of values into the database, make sure that there is no existing value.

For example, assume that the Sales figures for January are calculated by adding the values for each week in January:

```
January Sales = Week 1 Sales + Week 2 Sales + Week 3 Sales + Week 4 Sales
```

When you load Week 1 Sales, clear the database value for January Monthly Sales. If there is an existing value, Essbase performs the following calculation:

```
January Sales = Existing Value + Week 1 Sales + Week 2 Sales + Week 3 Sales
+ Week 4 Sales
```

You can also clear data from fields that are not part of the data load. For example, if a data source contains data for January, February, and March, and you want to load only the March data, you can clear January and February data.

➤ To clear existing values, see "Clearing Existing Data Values" in the *Oracle Essbase Administration Services Online Help*.

Note:

If you are using transparent partitions, clear the values using the steps that you use to clear data from a local database.

Replacing All Data

This section applies to loading data into an aggregate storage database only. If you are loading data into a block storage database or performing a dimension build, skip this section.

In an aggregate storage database, Essbase can remove all of the data in the database or all of the data in each incremental data slice in a database, and replace the data with the contents of a specified data load buffer. This functionality is useful when working with data sets that are small enough to reload completely, or when working with data that can be separated into large, static data sets that are never updated and small, volatile data sets in which you need to track changes.

To replace all data, see "Replacing Database or Incremental Data Slice Contents" on page 965 and "Replacing the Contents of an Aggregate Storage Database" in the *Oracle Essbase Administration Services Online Help*.

Scaling Data Values

This section is for data load only. If you are performing a dimension build, skip this section.

You can scale data values if the values of the data source are not in the same scale as the values of the database.

For example, assume the real value of sales is $5,460. If the Sales data source tracks the values in hundreds, the value is 54.6. If the Essbase database tracks the real value, you must multiply the value coming in from the Sales data source (54.6) by 100 to have the value display correctly in the Essbase database (as 5460).

➤ To scale data values, see "Scaling Data Values" in the *Oracle Essbase Administration Services Online Help*.

Flipping Field Signs

This section is for data load only. If you are performing a dimension build, skip this section.

You can reverse, or flip, the value of a data field by flipping its sign. Sign flips are based on the UDAs of the outline. When loading data into the accounts dimension, for example, you can specify that any record whose accounts member has a UDA of Expense change from a plus sign to a minus sign. See "Creating UDAs" on page 157.

➤ To reverse a field sign, see “Flipping Signs” in the *Oracle Essbase Administration Services Online Help*.

19 Performing and Debugging Data Loads or Dimension Builds

In This Chapter

Also see:

- Chapter 16, "Understanding Data Loading and Dimension Building"
- Chapter 17, "Working with Rules Files"

Prerequisites for Data Loads and Dimension Builds

You can load data or members from one or more external data sources to an Essbase Server. You can load data without updating the outline, update the outline without loading data, or load data and build dimensions simultaneously. Before you load data or build dimensions, ensure that the following items are in place:

- An Essbase database
- A connection to the appropriate Essbase Server
- One or more valid data sources

  See "Data Sources" on page 258.
- A rules files, if you are using one

  See "Rules Files" on page 263.
- If you are not using a rules file, a data source correctly formatted for free-form data loading

  See "Data Sources that Do Not Need a Rules File" on page 264.

Performing Data Loads or Dimension Builds

When you use Administration Services to perform a data load or dimension build, you can execute the load or build in the background so that you can continue working during the load or build. You can then check the status of the background process to see when the load or build has completed. See "Performing a Data Load or Dimension Build" in the *Oracle Essbase Administration Services Online Help*.

If you are using multiple data sources in a dimension build, to reduce total processing time you can perform a deferred-restructure dimension build. See "Performing Deferred-Restructure Dimension Builds" on page 299.

Note:

If you are loading data into a transparent partition, follow the same steps as for loading data into a local database.

➤ To load data or build dimensions, use a tool:

Tool	Topic	Location
Administration Services	Performing a Data Load or Dimension Build	*Oracle Essbase Administration Services Online Help*
MaxL	For data loading: **import data** For dimension building: **import dimensions**	*Oracle Essbase Technical Reference*
ESSCMD	For data loading: IMPORT For dimension building: BUILDDIM	*Oracle Essbase Technical Reference*

Stopping Data Loads or Dimension Builds

You can stop a data load or dimension build before it completes. You should not stop a data load or dimension build unless it is necessary. If a data load or dimension build process is terminated, Essbase displays the file name as partially loaded.

If you initiate a data load or dimension build from a client and terminate the data load or dimension build from the server, it could take time before the client responds to the termination request. Because Essbase reads the source file until all source data is read, the amount of time depends on the file size and the amount of source data that Essbase has processed. If the process is terminated from the computer that initiated it, termination is immediate.

Note:

If you are adding to or subtracting from data values during a data load to a block storage database, use the Committed Isolation Level setting, if possible. If the data load is terminated, this setting rolls the data load back to its previous state. See "Understanding Isolation Levels" on page 779. If you stop a data load that is adding to or subtracting from data values, see "Recovering from an Essbase Server Crash" on page 304.

➤ To stop a data load or dimension build before it completes, use a tool:

Tool	Topic	Location
Administration Services	Disconnecting User Sessions and Requests	*Oracle Essbase Administration Services Online Help*
MaxL	**alter system kill request**	*Oracle Essbase Technical Reference*

Tips for Loading Data and Building Dimensions

This section contains topics to help you load data and build dimensions.

Performing Deferred-Restructure Dimension Builds

Skip this section if you are loading data only or are using a single data source for a dimension build.

By default, each time you make changes to an outline, Essbase considers the type of change and restructures the database if needed. Restructuring the database rebuilds the database, which takes time, requires more disk space for its process, and can cause file defragmentation. For information about the types of changes and types of restructuring that can be required, see the following topics:

- For block storage outlines, see "Optimizing Database Restructuring" on page 841.
- For aggregate storage outlines, see "Aggregate Storage Database Restructuring" on page 994.

Deferred-restructure dimension builds, also known as incremental dimension builds, read multiple data sources for dimension builds and delay restructuring until all data sources have been processed. The following methods for performing deferred-restructure builds use different approaches to specifying the action:

- Administration Services enables you to list all data sources in a single dialog box.

  When the files listed are all dimension builds, a deferred-restructure dimension build option is available. Selecting this option delays restructuring until all sources have been processed. The outline is validated after each dimension build is processed. See "Performing a Data Load or Dimension Build" in the *Oracle Essbase Administration Services Online Help*.

- MaxL enables you to include all of the data sources within one **import database** statement.

  You can control whether outline validation is performed for each file. You must enable outline validation for the last file. See "import database" and "import dimension" in the MaxL section of the *Oracle Essbase Technical Reference*.

- ESSCMD requires several commands:
  - BEGINCBUILDDIM, to indicate that a deferred-restructure dimension build is to be performed

- ❍ INCBUILDDIM for each data source, to indicate that the data sources to be included in the dimension build
- ❍ ENDBUILDDIM, to trigger restructuring, if needed

In all cases, the data sources are processed in the order in which they are listed.

Note:

MaxL and ESSCMD enable you to enforce or suppress outline verification for each file. To ensure a valid outline, ensure that the last build verifies the outline. Deferred-restructure dimension builds in Administration Services verify the outline after each data source is processed.

Determining Where to Load Data

Skip this section if you are building dimensions or working with an aggregate storage database.

If you load data into a parent member, when you calculate the database, the consolidation of the children's data values can overwrite the parent data value. To prevent overwriting:

- If possible, do not load data directly into a parent.
- If you must load data into a parent member, ensure that Essbase knows not to consolidate #MISSING values from the children of the parent into the parent.

➤ To set the consolidation, use a tool:

Tool	Topic	Location
Administration Services	Aggregating Missing Values During Calculation	*Oracle Essbase Administration Services Online Help*
Calculation Script	SET AGGMISSG	*Oracle Essbase Technical Reference*
MaxL	**alter database**	*Oracle Essbase Technical Reference*
ESSCMD	SETDBSTATEITEM	*Oracle Essbase Technical Reference*

The methods in this table work only if the child values are empty (#MISSING). If the children have data values, the data values overwrite the data values of the parent. See "Consolidating #MISSING Values" on page 894.

Note:

You cannot load data into Dynamic Calc, Dynamic Calc and Store, or attribute members. For example, if Year is a Dynamic Calc member, you cannot load data into it. Instead, load data into Qtr1, Qtr2, Qtr3, and Qtr4, which are not Dynamic Calc members.

Dealing with Missing Fields in a Data Source

Each record in the data source must have the same number of data value fields to perform a data load. If data values are missing, the data load processes incorrectly.

For example, the following file is invalid, because there is no value under Apr:

```
Actual Ohio Sales Cola
Jan     Feb    Mar    Apr
10      15     20
```

To fix the file, insert #MISSING or #MI into the missing field:

```
Actual Ohio Sales Cola
Jan     Feb    Mar    Apr
10      15     20     #MI
```

See "Replacing an Empty Field with Text" on page 292.

Note:

If a dimension field or member field is missing, Essbase uses the value that it used previously for that dimension or member field. See "Missing Dimension or Member Fields" on page 307.

If a rules file has extra blank fields, join the empty fields with the field next to them. See "Joining Fields" on page 290.

In aggregate storage databases, values can be loaded only to level 0 cells. Specifying #MISSING or #MI as a value in the data source removes the associated cell if it is present in the database. For information about data load differences for aggregate storage databases, see "Preparing Aggregate Storage Databases" on page 954.

Loading a Subset of Records from a Data Source

You can load a subset of records in a data source during a data load or a dimension build. For example, you can load records 250 to 500 without loading the other records of the data source.

➤ To load a subset of records:

1. **Using a text-editing tool, number the records in the data source.**
2. **Set the rules file to ignore the column containing the record number.**

   See "Ignoring Fields" on page 289.
3. **Define a rejection criterion that rejects all records except those that you want to load.**

   For example, reject all records for which the ignored column is fewer than 250 or greater than 500. See "Rejecting Records" on page 285.

Note:

You cannot reject more records than the error log can hold. By default, the limit is 1000. You can change the limit by setting DATAERRORLIMIT in the `essbase.cfg` file. See the *Oracle Essbase Technical Reference*.

Saving and Reusing Data Load Options

If you perform data loads and dimension builds through Administration Services, and you use the same data source files and rules files each time, you can save the source and rules file options specified in the Data Load dialog box and reuse them later.

By default, the file options are saved in the *ARBORPATH* directory on the client in an XML file that you name. To use a file of saved options, you can open the file. The saved files and their options are added to the list in the currently open Data Load dialog box.

Tip:

You can use the data load options file as a template; after opening a data load options file, you can modify any specifications.

See "Performing a Data Load or Dimension Build" in the *Oracle Essbase Administration Services Online Help*.

Debugging Data Loads and Dimension Builds

If a data source does not correctly load into Essbase Server, ensure that you are connected to the appropriate application and database and that you are loading the correct data source.

If you still encounter problems, see the following topics. After you correct the problems, you can reload the records that did not load by reloading the error log. See "Loading Dimension Build and Data Load Error Logs" on page 755.

Verifying that Essbase Server Is Available

To help identify that the problem is with Essbase and not with the server or network, try to access the server without using Essbase. Check whether:

- The server is running.

  Try connecting to the server without using Essbase. If you cannot, check with your system administrator.
- Essbase Server is running.

  Check with your Essbase administrator.
- The client is connected to the server.

  Try connecting to the server from the client without using Essbase.

Verifying that the Data Source Is Available

If Essbase cannot open the data source that you want to load, ensure that the following conditions are true:

- The data source is open (for example, is someone editing the data source?).

  Essbase can load only data sources that are not locked by another user or application.
- The data source has the correct file extension.

  Text files must have a `.txt` extension; rules files must have an `.rul` extension.
- The data source name and the path name are correct.

  Check for misspellings.
- The data source is in the specified location.

  Ensure that no one has moved or deleted the data source.
- If you are using a SQL data source:
  - The connection information (such as the user name, password, and database name) is correct.
  - You can connect to the SQL data source without using Essbase.

Checking Error Logs

If a data load or dimension build fails, the error log can be a valuable debugging tool. See "Understanding and Viewing Dimension Build and Data Load Error Logs" on page 754.

If there is no error log, check whether the following conditions exist:

- The person running the data load set up an error log.

  By default, when using a rules file, Essbase creates an error log.
- The data source and Essbase Server are available.

  See "Verifying that Essbase Server Is Available" on page 302 and "Verifying that the Data Source Is Available" on page 302
- Essbase Server crashed during the data load.

  If so, you probably received a timeout error on the client. See "Recovering from an Essbase Server Crash" on page 304.
- The application log exists.

  See "Essbase Server and Application Logs" on page 730.

If the error log exists but is empty, Essbase does not think that an error occurred during loading. Check whether the following conditions exist:

- The rules file contains selection or rejection criteria that rejected every record in the data source.

  See "Selecting Records" on page 285 and "Rejecting Records" on page 285.
- The rules file validates properly.

  See "Requirements for Valid Data Load Rules Files" on page 281 and "Requirements for Valid Dimension Build Rules Files" on page 282.

Recovering from an Essbase Server Crash

If the server crashes while you are loading data, Essbase sends you a timeout error. The recovery procedures that you must perform depend on the type of load you are performing and the Isolation Level database transaction setting (see "Understanding Isolation Levels" on page 779):

- If you are overwriting the values of the data source, reload the data source when the server is running again.
- If you are adding to or subtracting from existing values in the data source and the Isolation Level setting is:
    - Committed, reload the data source when the server is running again.
    - Uncommitted, determine how much data Essbase loaded before the crash:
        1. Compare the values of the data source with the values of the database.
        2. If the values that you are adding to or subtracting from have not changed, reload the data source.
        3. If the values that you are adding to or subtracting from have changed, clear the values that loaded and reload the previous data sources. If, for example, you derive monthly sales figures by adding the sales figures for each week as they are loaded, clear the sales figures from the database and reload the sales figures for each week up to the current week.

Resolving Problems with Data Loaded Incorrectly

If the data source loads without error, but the data in the database is wrong, check whether the following conditions exist:

- You loaded the correct data source.

  If so, check the data source again to make sure that it contains the correct values.
- There are blank fields in the data source.

  You must insert #MI or #MISSING into a data field that has no value. Otherwise, the data source may not load correctly. To replace a blank field with #MI or #MISSING using a rules file, see "Replacing an Empty Field with Text" on page 292.
- The data source is formatted correctly.
    - All ranges are set up properly.
    - The data is clean. For example, as it processes the data source, Essbase recognizes member names and knows the dimensions they belong to. If a data source record inadvertently includes a member from a dimension for which there is a member named in the header record, the new member name replaces the header record member for that dimension. In the following example data source, Essbase recognizes Florida as a member of the Market dimension. The values in the last four records are interpreted as Florida values instead of Texas values.

```
Jan     Actual     Texas     Sales
 "100-10"     51.7
 "100-20"     102.5
 "100-20"     335.0
 Florida      96.7
 "200-20"     276.0
 "200-20"     113.1
 "200-10"     167.0
```

- There are any implicitly shared members (when a parent and child share the same data value) of which you were unaware.

  This situation occurs if a parent has only one child or only one child rolls up into the parent.

  See "Understanding Implied Sharing" on page 152.

- You added incoming data to existing data instead of replacing incoming data with existing data.

  See "Adding to and Subtracting from Existing Values" on page 293.

- You selected or rejected any records that you did not intend to select or reject.

  See "Selecting Records" on page 285 and "Rejecting Records" on page 285.

- The sign is reversed (for example, a minus sign instead of a plus sign) and whether you performed sign flips on any UDAs.

  See "Flipping Field Signs" on page 295.

- You cleared data combinations that you did not intend to clear.

  See "Clearing Existing Data Values" on page 294.

- You scaled the incoming values incorrectly.

  See "Scaling Data Values" on page 295.

- All member and alias names are fewer than 79 characters long.

Note:

You can check data by exporting it, by running a report on it, or by using a spreadsheet. If exporting or running reports, see Chapter 33, "Developing Report Scripts" and Appendix E, "Using ESSCMD." If using a spreadsheet, see *Oracle Essbase Spreadsheet Add-in User's Guide*.

Creating Rejection Criteria for End of File Markers

A SQL data source may have an end of file marker made up of special characters that cause a data load or dimension build to fail. To fix this problem, define a rejection criterion to reject the problem record.

1. Find the end of file marker in the SQL data source.
2. Determine how to search for the end of file marker using the Essbase search command.

   This task may be difficult, because the end of file marker may be composed of one or more special characters.

See “Ignoring Fields Based on String Matches” in the *Oracle Essbase Administration Services Online Help*.

3. Define a rejection criterion that rejects the end of file marker.

   See “Rejecting Records” in the *Oracle Essbase Administration Services Online Help*.

Understanding How Essbase Processes a Rules File

Sometimes, you can track down problems with dimension builds by understanding how Essbase initializes the rules file and processes the data source.

Essbase performs the following steps to initialize a rules file:

1. Validates the rules file against the associated outline.
2. Validates the dimensions. This process includes ensuring that the build method and field types are compatible and that each dimension name is unique. Member names must be either unique or shared.
3. Adds new dimensions defined in the rules file to the outline.
4. Reads header records specified in the data source.

Then Essbase performs the following operations on each record of the data source during a data load or dimension build:

1. Sets the file delimiters for all records.
2. Applies field operations to the data in the order in which the operations are defined in the rules file.

   Field operations include joins, moves, splits, and creating fields using text and joins. To see the order in which field operations are defined in the rules file, see “Performing Operations on Fields” on page 288.
3. Essbase applies all properties for each field, applying all properties to field1 before proceeding to field2. Essbase applies field properties in the following order:
   a. Ignores fields set to be ignored during data load
   b. Ignores fields set to be ignored during dimension build
   c. Flags the data field
   d. Applies field names
   e. Applies field generations
   f. Performs all replaces in the order in which they are defined in the rules file
   g. Drops leading and trailing spaces
   h. Converts spaces to underscores
   i. Applies suffix and prefix operations
   j. Scales data values
   k. Converts text to lowercase

    l. Converts text to uppercase

4. Adds members, or member information, or both, to the outline
5. If you chose to skip lines, Essbase skips the number of lines that you specified; otherwise, Essbase proceeds to the first record.
6. Essbase performs selection or rejection criteria in the order in which the criteria are defined in the rules file. Essbase loads or rejects individual records of the data source based on the specified criteria.

Understanding How Essbase Processes Missing or Invalid Fields During a Data Load

The following sections describe how Essbase processes invalid fields during a data load.

Missing Dimension or Member Fields

If a dimension or member field is missing, Essbase uses the value that it used previously for that dimension or member field. If there is no previous value, Essbase aborts the data load.

For example, when you load the following file into the Sample.Basic database, Essbase maps the Ohio member field into the Market dimension for all records, including the records that have Root Beer and Diet Cola in the Product dimension.

```
Jan Sales Actual Ohio
                 Cola          25
                 "Root Beer"   50
                 "Diet Cola"   19
```

Essbase stops the data load if no prior record contains a value for the missing member field. For example, if you try to load the following file into the Sample.Basic database, the data load stops, because the Market dimension (Ohio, in the previous example) is not specified.

```
Jan Sales Actual
           Cola          25
           "Root Beer"   50
           "Diet Cola"   19
```

For information on restarting the load, see "Loading Dimension Build and Data Load Error Logs" on page 755.

Unknown Member Fields

If you are performing a data load and Essbase encounters an unknown member name, Essbase rejects the entire record. If there is a prior record with a member name for the missing member field, Essbase continues to the next record. If there is no prior record, the data load stops. For example, when you load the following file into the Sample.Basic database, Essbase rejects the record containing Ginger Ale because it is not a valid member name. Essbase loads the records containing Cola, Root Beer, and Cream Soda. If Ginger Ale were in the first record, however, the data load would stop.

```
Jan, Sales, Actual
Ohio    Cola          2
        "Root Beer"   12
        "Ginger Ale"  15
        "Cream Soda"  11
```

Note:

If you are performing a dimension build, you can add the new member to the database. See "Performing Data Loads or Dimension Builds" on page 297.

For information on restarting the load, see "Loading Dimension Build and Data Load Error Logs" on page 755.

Invalid Data Fields

If you are performing a data load, when Essbase encounters an invalid data field, it stops the data load. Essbase loads all fields read before the invalid field into the database, resulting in a partial data load. For example, in the following file, Essbase stops the data load when it encounters the 15- data value. Essbase loads the Jan and Feb Sales records but not the Mar and Apr Sales records.

```
East Cola   Actual
Sales       Jan     $10
            Feb     $21
            Mar     $15-
            Apr     $16
```

For information on continuing the load, see "Loading Dimension Build and Data Load Error Logs" on page 755.

20 Understanding Advanced Dimension Building Concepts

In This Chapter

Understanding Build Methods

The build method that you select depends on the type of data in the data source and determines the algorithm that Essbase uses to add, change, or remove dimensions, members, and aliases in the outline.

Use the guidelines in Table 42 to select the appropriate build method for the data source:

Table 42 Build Method Guidelines

Type of Data in Each Record	Examples	Desired Operation	Build Method*	Field Type Information
Top-down data Each record specifies the parent's name, the child's name, the children of that child, and so on.	Year, Quarter, Month	Modify the properties of existing dimensions and members	Generation references	The generation number for each field.
Bottom-up data Each record specifies the name of the member, the name of its parent, the name of its parent's parent, and so forth.	Month, Quarter, Year	● Create shared members that roll up into different generations ● Modify the properties of existing dimensions and members	Level references	The level number for each field.

Type of Data in Each Record	Examples	Desired Operation	Build Method*	Field Type Information
Parent followed by its child Each record specifies the name of the parent and the name of the new child member, in that order, although they can specify other information as well.	Cola, Diet Cola	● Create shared members that roll up into different generations ● Share non-level 0 members ● Modify properties of existing dimensions and members	Parent-child references	Whether a field is parent or child. The field number is 0.
A list of new members Each data source lists new members; the data source does not specify where in the outline the members belong. Essbase provides algorithms that determine where to add these members.	Jan, Feb, Mar, April	Add all members as children of an existing parent (possibly a "dummy" parent)	Add as child of the specified parent	
	800-10, 800-20	Add all members at the end of the dimension	Add as sibling at the lowest level	
	800-10, 800-20	Add each new member to the dimension that contains similar members	Add as sibling to a member with a matching string	
A list of base dimension members and their attributes	Cola 16oz Can, Root Beer 14oz Bottle	Add members to an attribute dimension and associate the added members with the appropriate members of the base dimension	Generation, level, or parent-child references, depending on the organization of the source data	The number for each field. The number is either the generation or level number of the associated member of the base dimension or zero.

*Using a level references build, you cannot create an alias that has the same name as its member. This restriction does not apply if you use other build methods, including the generation references build method.

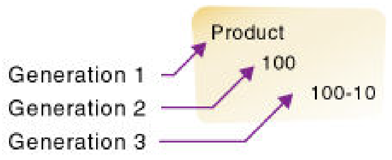
Using Generation References

Top-down data sources are organized left to right from the highest level to the lowest level. Each record begins with the most general information and progresses to the most specific information. The name of the new member is at the end of the record. When using a top-down data source, use the generation references build method. In the rules file, specify the generation number and the field type of each field of the data source.

Essbase numbers members within a dimension according to the hierarchical position of the member within the dimension. The numbers are called *generation references*. A dimension is always generation 1. All members at the same branch in a dimension are called a *generation*. Generations are numbered top-down according to their position relative to the dimension; that is, relative to dimension 1.

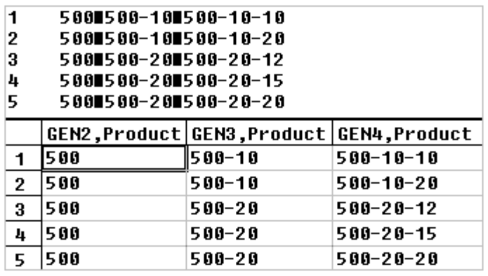
For example, as illustrated in Figure 64, the Product dimension in the Sample.Basic database is generation 1. Product has a 100 member, which is generation 2. 100 has members, such as 100-10, which are generation 3. To use the generation references build method, specify the generation reference number in the rules file.

Figure 64 Generations



The top half of Figure 65 shows a top-down data source (GENREF.TXT). The data source is used to build the Product dimension. The bottom half shows the rules file for the data source (GENREF.RUL). The rules file specifies the generation number for each field in the data source. See "Setting Field Type Information" on page 277.

Figure 65 Rules File for Generation Build

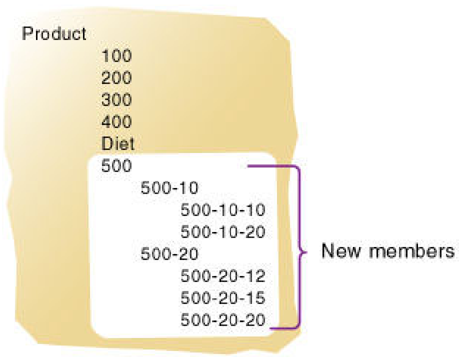


```
500■500-10■500-10-10
500■500-10■500-10-20
500■500-20■500-20-12
500■500-20■500-20-15
500■500-20■500-20-20
```

	GEN2,Product	GEN3,Product	GEN4,Product
1	500	500-10	500-10-10
2	500	500-10	500-10-20
3	500	500-20	500-20-12
4	500	500-20	500-20-15
5	500	500-20	500-20-20

Figure 66 shows the tree that Essbase builds from the GENREF.TXT data source and GENREF.RUL rules file:

Figure 66 Generation References



Dealing with Empty Fields

When you use the generation references build method, you can choose to process null values. Null processing specifies what actions Essbase takes when it encounters empty fields, also known as null fields, in the data source.

If null processing is not enabled, Essbase rejects all records with null values and writes an error to the error log.

If null processing is enabled, Essbase processes nulls as in the following ways:

- **Missing field:** If the null occurs where Essbase expects a GENERATION field, Essbase promotes the next GENERATION field to replace the missing field.

  In the following example, there is no field in the GEN3,Products column:

```
GEN2,Products   GEN3,Products   GEN4,Products
100                             100-10a
```

  When Essbase reads the record, it promotes the GEN4 field (100-10a) to GEN3, as if the data source looked like the following example:

```
GEN2,Products   GEN3,Products   GEN4,Products
100             100-10a
```

- **Missing field before secondary field:** If a null occurs directly before a secondary field, Essbase ignores the secondary field. (Secondary field types are alias, property, formula, duplicate generation, duplicate generation alias, currency name, currency category, attribute parent, UDA, and name of an attribute dimension.)

  In the following example, there is no field in the GEN2, Products or the ALIAS2,Products column:

```
GEN2,Products   ALIAS2,Products   GEN3,Products   GEN4,Products
                Cola              100-10          100-10a
```

  When Essbase reads the record, it ignores the ALIAS2 field and promotes the GEN3 field (100-10) to GEN2 and the GEN4 field (100-10a) to GEN3, as if the data source looked like the following example:

```
GEN2,Products   ALIAS2,Products   GEN3,Products   GEN4,Products
100-10          Cola              100-10a
```

- **Missing secondary field:** If the null occurs where Essbase expects a secondary field, Essbase ignores the secondary null field and continues loading.

  In the following example, there is no field in the ALIAS2, Products column:

```
GEN2,Products   ALIAS2,Products   GEN3,Products   GEN4,Products
100                               100-10          100-10a
```

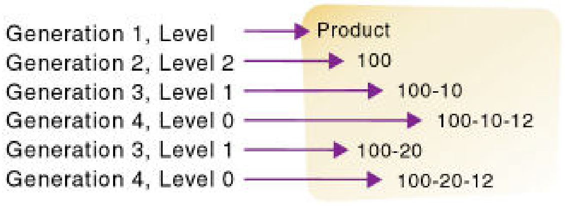
  When Essbase reads the record, it ignores the ALIAS2 field and loads the other fields.

Using Level References

In a bottom-up data source, each record defines a single member of a dimension. The definition begins with the most specific information about the member and provides progressively more general information. A typical record specifies the name of the new member, then the name of its parent, then its parent's parent, and so forth.

Levels are defined from a bottom-up hierarchical structure. For example, in the outline in Figure 67, the lowest-level members are at the bottoms of the branches of the Product dimension.

Figure 67 Generation and Level Numbers



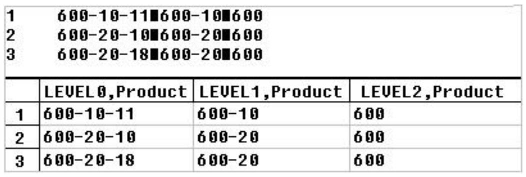
To build the outline in Figure 67, you can use the following bottom-up data source:

```
100-10-12 100-10 100
100-20-12 100-20 100
```

In a level reference build, the lowest-level members are sequenced left to right. Level 0 members are in the first field, level 1 members are in the second field, and so on. This organization is the opposite of how data is presented for generation references (top-down).

In the following example, the rules file uses the level reference build method to add members to the Product dimension of the Sample.Basic database. The rules file specifies the level number and the field type for each field of the data source (see "Setting Field Type Information" on page 277). The first column of the data source contains new members (600-10-11, 600-20-10, and 600-20-18). The second column contains the parents of the new members (600-10 and 600-20), and the third column contains parents of the parents (600).

Figure 68 Rules File for Level Build

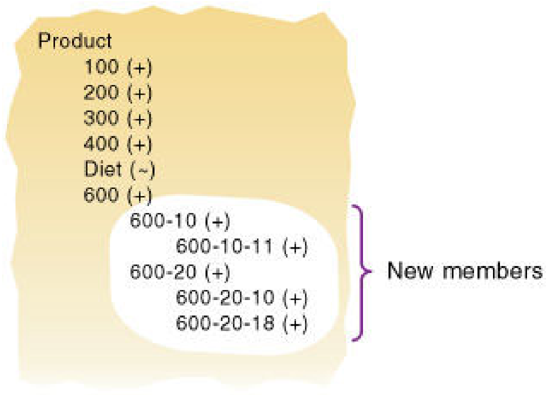


```
1    600-10-11■600-10■600
2    600-20-10■600-20■600
3    600-20-18■600-20■600
```

	LEVEL0,Product	LEVEL1,Product	LEVEL2,Product
1	600-10-11	600-10	600
2	600-20-10	600-20	600
3	600-20-18	600-20	600

For example, to build the tree in Figure 69, use Figure 68 to set up the data source (LEVEL.TXT) and the rules file (LEVEL.RUL).

Figure 69 Levels



Dealing with Empty Fields

When you use the level references build method, you can choose to process null values. Null processing specifies what actions Essbase takes when it encounters empty fields, also know as null fields, in the data source.

If null processing is not enabled, Essbase rejects all records with null values and writes an error to the error log.

If null processing is enabled, Essbase processes nulls in the following ways:

- **Missing field:** If a null occurs where Essbase expects a LEVEL field, Essbase promotes the next LEVEL field to replace the missing field.

  In the following example, there is no field in the LEVEL0, Products column:

  ```
  LEVEL0,Products    LEVEL1,Products    LEVEL2,Products
                     100-10             100
  ```

  When Essbase reads the record, it promotes the LEVEL1 field (100-10) to LEVEL0 and the LEVEL2 field (100) to LEVEL1, as if the data source looked like the following example:

  ```
  LEVEL0,Products    LEVEL1,Products    LEVEL2,Products
  100-10             100
  ```

- **Missing field before a secondary field:** If a null occurs directly before a secondary field, Essbase ignores the secondary field. (Secondary field options are alias, property, formula, duplicate level, duplicate level alias, currency name, currency category, attribute parent, UDA, and a name of an attribute dimension.)

  In the following example, there is no field in the LEVEL0, Products column:

  ```
  LEVEL0,Products    ALIAS0,Products    LEVEL1,Products    LEVEL2,Products
                     Cola               100-10             100
  ```

  When Essbase reads the record, it ignores the ALIAS0 field and promotes the LEVEL1 field (100-10) to LEVEL0 and the LEVEL2 field (100) to LEVEL1, as if the data source looked like the following example:

  ```
  LEVEL0,Products    ALIAS0,Products    LEVEL1,Products    LEVEL2,Products
  100-10              Cola                100
  ```

- **Missing secondary field:** If a null occurs where Essbase expects a secondary field, Essbase ignores the secondary null field and continues loading.

  In the following example, there is no field in the ALIAS0, Products column:

  ```
  LEVEL0,Products    ALIAS0,Products    LEVEL1,Products    LEVEL2,Products
  100-10a            100-10             100
  ```

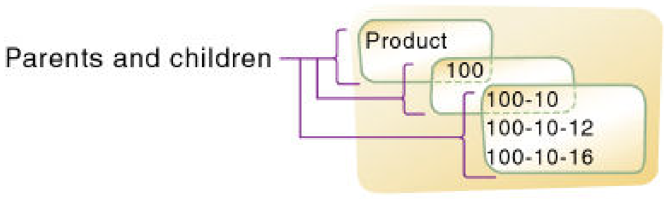
  When Essbase reads the record, it ignores the ALIAS0 field and loads the other fields.

Using Parent-Child References

Use the parent-child references build method when every record of the data source specifies the name of a new member and the name of the parent to which you want to add the new member.

Members in a database exist in a parent-child relationship. Figure 70 shows part of the Product dimension with its parent and children relationships identified.

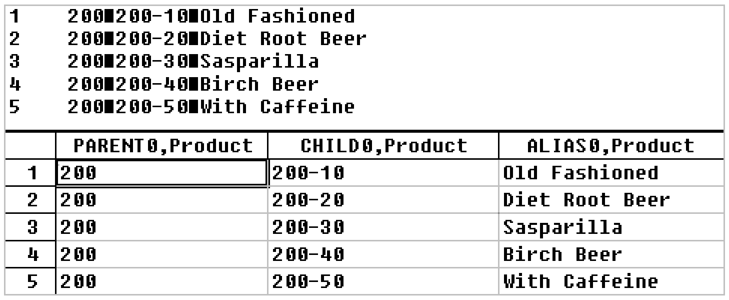
Figure 70 Parents and Children



A parent-child data source must contain at least two columns: a parent column and a child column, in that order. The data source can include columns with other information (for example, the alias, the attributes, or the properties of the new member). A record within a parent-child data source cannot specify more than one parent or more than one child and cannot reverse the order of the parent and child columns.

In a parent-child build, the rules file specifies which column is the parent and which column is the child. For example, the top half of Figure 71 shows a data source (`PARCHIL.TXT`), in which each record specifies the name of a parent and the name of its child, in that order. The bottom half of the figure shows the rules file (`PARCHIL.RUL`) that specifies which column is the parent and which column is the child. Additionally, this example associates aliases with the child field.

Figure 71 Rules Files for Parent-Child Build

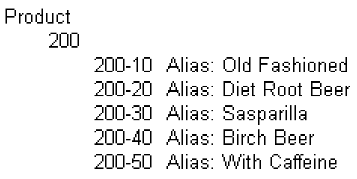


```
1    200■200-10■Old Fashioned
2    200■200-20■Diet Root Beer
3    200■200-30■Sasparilla
4    200■200-40■Birch Beer
5    200■200-50■With Caffeine
```

	PARENT0,Product	CHILD0,Product	ALIAS0,Product
1	200	200-10	Old Fashioned
2	200	200-20	Diet Root Beer
3	200	200-30	Sasparilla
4	200	200-40	Birch Beer
5	200	200-50	With Caffeine

Figure 72 shows the tree that Essbase builds from this data source and rules file.

Figure 72 Parents and Children



```
Product
    200
        200-10  Alias: Old Fashioned
        200-20  Alias: Diet Root Beer
        200-30  Alias: Sasparilla
        200-40  Alias: Birch Beer
        200-50  Alias: With Caffeine
```

Note:

For duplicate member situations, the parent field must contain the qualified member name. See "Building Qualified Member Names Through the Rules File" on page 340.

Adding a List of New Members

If a data source consists of a list of new members and does not specify their ancestors, Essbase must decide where in the outline to add them. Essbase provides the following build methods for this type of data source:

- Add each new member as a sibling of the existing member whose text most closely matches its own.

  See "Adding Members Based On String Matches" on page 316.
- Add each new member as a sibling of the lowest-level existing member.

  See "Adding Members as Siblings of the Lowest Level" on page 317.
- Add all new members as children of a specified parent (generally a "dummy" parent).

  See "Adding Members to a Specified Parent" on page 318.

After Essbase adds all new members to the outline, it may be necessary to move the new members into their correct positions using Outline Editor. See "Positioning Dimensions and Members" on page 130.

Note:

Essbase does not support concurrent attribute association with the Add as build methods.

Adding Members Based On String Matches

You can add new members from a data source to an existing dimension by matching strings with existing members. When Essbase encounters a new member in a data source, it scans the outline for a member name with similar text and adds the new member as a sibling of the member with the closest string match.

For example, the data source in Figure 73 (`SIBSTR.TXT`) contains two new members (100-11 and 200-22) to add to the Product dimension in the Sample.Basic database. The new members are similar to strings in the Product dimension: they contain three digits, one dash, and two digits.

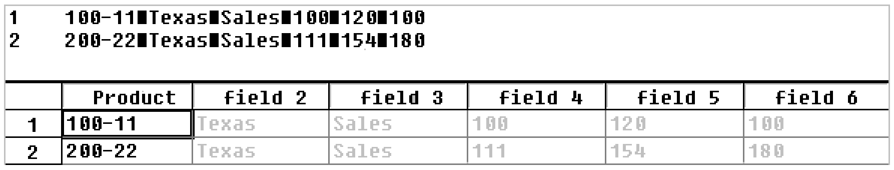
To add the example members to the database, set the values in Table 43 in the rules file:

Table 43 Example of Adding Members Using String Matches

Field	Value	See
Field 1 (Product)	● Do not select a field type for the field	"Setting Field Type Information" on page 277

Field	Value	See
	● Set the dimension for the field to Product (field 1 is displayed as Product, as shown in Figure 72)	
Fields 2 through 6	Ignore the fields	“Ignoring Fields” on page 289
Product dimension	Select the “Add as sibling of matching string” build method	“Selecting a Build Method” on page 274

Figure 73 Rules File Fields Set to Add Members as Siblings with String Matches

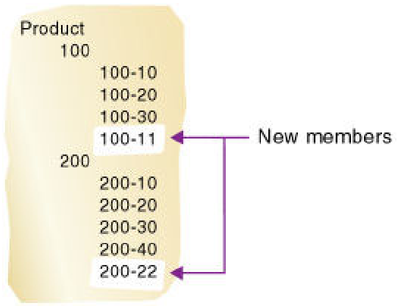


```
1    100-11▮Texas▮Sales▮100▮120▮100
2    200-22▮Texas▮Sales▮111▮154▮180
```

	Product	field 2	field 3	field 4	field 5	field 6
1	100-11	Texas	Sales	100	120	100
2	200-22	Texas	Sales	111	154	180

Figure 74 shows the tree that Essbase builds from this data source and rules file.

Figure 74 Tree for Adding Members as Siblings with String Matches



Adding Members as Siblings of the Lowest Level

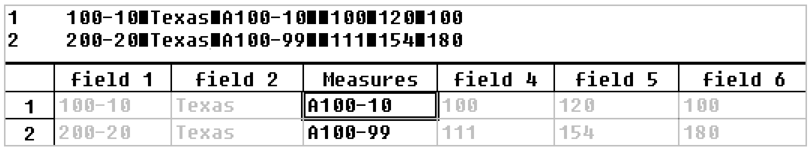
You can add new members from a data source as siblings of members that reside at the lowest level of a dimension—at the level 0 branch. When Essbase encounters a new member in a data source, it scans the outline for the level 0 branch of members and adds the new member as a sibling of these members.

Note:

If the outline contains more than one group of members at this level, Essbase adds the new member to the first group of members that it encounters.

For example, the data source (`SIBLOW.TXT`) and the rules file (`SIBLOW.RUL`) in Figure 75 contain new members (A100-10 and A100-99) to add to the Measures dimension of the Sample.Basic database.

Figure 75 Rules File Fields Set to Add Members as Siblings of the Lowest Level



```
1     100-10■Texas■A100-10■■100■120■100
2     200-20■Texas■A100-99■■111■154■180
```

	field 1	field 2	Measures	field 4	field 5	field 6
1	100-10	Texas	A100-10	100	120	100
2	200-20	Texas	A100-99	111	154	180

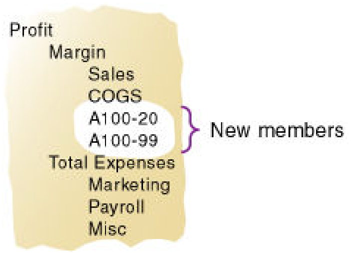
To add the example members dynamically to the database, set the values shown in Table 44 in the rules file:

Table 44 Example of Adding Members as Siblings of the Lowest Level

Field	Value	See
Field 3 (Measures)	● Do not select a field type for the field ● Set the dimension for the field to Measures (field 3 is displayed as Measures, as shown in Figure 75)	"Setting Field Type Information" on page 277
Fields 1, 2, 4, 5, and 6	Ignore the fields	"Ignoring Fields" on page 289
Measures dimension	Select the "Add as sibling of lowest level" build method	"Selecting a Build Method" on page 274

Figure 76 shows the tree that Essbase builds from this data source and rules file.

Figure 76 Tree for Adding Members as Siblings of the Lowest Level



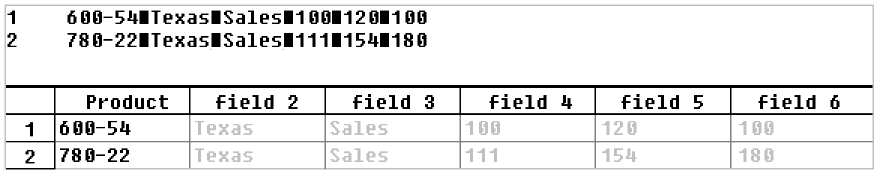
Adding Members to a Specified Parent

You can add all new members as children of a specified parent, generally a "dummy" parent. After Essbase adds all new members to the outline, review the added members and move or delete them in Outline Editor.

When Essbase encounters a new member in the data source, it adds the new member as a child of the parent that you define. The parent must be part of the outline before you start the dimension build.

For example, the data source in Figure 77 (SIBPAR.TXT) contains two new members (600-54 and 780-22) for the Product dimension (field 1). Assume that you previously added a member called NewProducts under the Products dimension.

Figure 77 Rules File Fields Set to Add Members as a Child of a Specified Parent



```
1     600-54■Texas■Sales■100■120■100
2     780-22■Texas■Sales■111■154■180
```

	Product	field 2	field 3	field 4	field 5	field 6
1	600-54	Texas	Sales	100	120	100
2	780-22	Texas	Sales	111	154	180

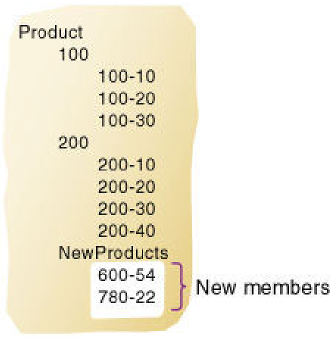
To add the example members to the database under the NewProducts member, set the values shown in Table 45 in the rules file:

Table 45 Example of Adding Members as a Child of a Specified Parent

Field	Value	See
Field 1 (Product)	● Do not select a field type for the field ● Set the dimension for the field to Product (field 1 is displayed as Product, as shown in Figure 77)	"Setting Field Type Information" on page 277
Fields 2 through 6	Ignore the fields	"Ignoring Fields" on page 289
Product dimension	Select the "Add as child of" build method	"Selecting a Build Method" on page 274 Enter `NewProducts` in the Add as Child of text box.

Figure 78 shows the tree that Essbase builds from this data source and rules file.

Figure 78 Tree for Adding Members as a Child of a Specified Parent



Building Attribute Dimensions and Associating Attributes

When a data source contains attribute information, you must use one or more rules files to build attribute dimensions and to associate attributes with members of their base dimensions.

You can use rules files to build attribute dimensions dynamically, to add and delete members, and to establish or change attribute associations.

Working with attributes involves the following operations:

- If the base dimension does not exist, you must build it.

- You must build the attribute dimension.
- You must associate members of the base dimension with members of the attribute dimension.

You can use any of the following approaches to perform these operations:

- Build the base and attribute dimensions and perform the associations all at once. Doing so, you use a single rules file to build the base dimension and one or more attribute dimensions to associate each attribute with the appropriate member of the base dimension. Because this approach uses a single rules file, it can be the most convenient. Use this approach if the base dimension does not exist and each source data record contains all attribute information for each member of the base dimension.
- Build the attribute dimension and perform the associations in one rules file. Assuming that the base dimension is built in a separate step or that the base dimension already exists, you can build an attribute dimension and associate the attributes with the members of the base dimension in one step. You need only to define the attribute associations in the rules file. See "Associating Attributes" on page 321.
- Build the attribute dimension and then perform the associations using separate rules files. Assuming that the base dimension is built in a separate step or that the base dimension already exists, you can build an attribute dimension and associate the attributes with the members of the base dimension in separate steps. Build the attribute dimension, and then associate the attribute members with members of the base dimension. Use this approach when you build numeric attribute dimensions that are multilevel or that have members that represent different-sized ranges.

The following sections describe how to build attribute dimensions.

Building Attribute Dimensions

Before you build attribute dimensions in a database, you must define the attribute member name formats for the outline. See "Setting Member Names in Attribute Dimensions" on page 170.

You can build attribute dimensions in one of the following ways:

- The same way in which you build standard dimensions.

  See"Process for Data Loading and Dimension Building" on page 257.
- At the same time as you associate attributes with members of the base dimension.

  See "Associating Attributes" on page 321.

Essbase does not support concurrent attribute association with the Add as build methods.

When you define the rules file for building attribute dimensions, specify the base dimension and the name of the attribute dimension file.

Associating Attributes

Whether you build the attribute dimension and associate the attribute members with the members of the base dimension in one step or in separate steps, define the fields as described in this section.

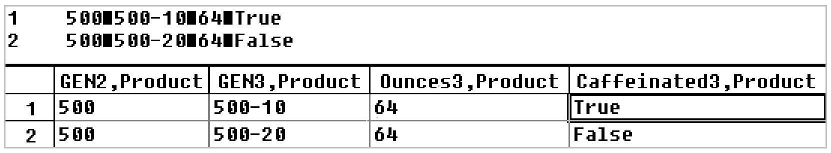
Note:

If you are working with a multilevel attribute dimension or with an attribute dimension of the type numeric, Boolean, or date, the rules file requires an additional field. See "Working with Multilevel Attribute Dimensions" on page 322.

Every record of the source data must include at least two columns: one for the member of the base dimension and one for the attribute value of the base dimension member. In the same source data record you can include additional columns for other attributes that you want to associate with the member of the base dimension. You must position the field for the member of the base dimension before any of the fields for the members of the attribute dimension.

Define the field type for the attribute dimension member as the name of the attribute dimension, use the generation or level number of the associated member of the base dimension, and specify the base dimension name. For example, as shown in the `ATTRPROD.RUL` file in Figure 79, the field definition Ounces3,Product specifies that the field contains members of the Ounces attribute dimension. Each member of this field is associated with the data field that is defined as the generation 3 member of the base dimension Product. Based on this field definition, Essbase associates the attribute 64 with the 500-10 member.

Figure 79 Rules File for Associating Attributes



```
1     500■500-10■64■True
2     500■500-20■64■False
```

	GEN2,Product	GEN3,Product	Ounces3,Product	Caffeinated3,Product
1	500	500-10	64	True
2	500	500-20	64	False

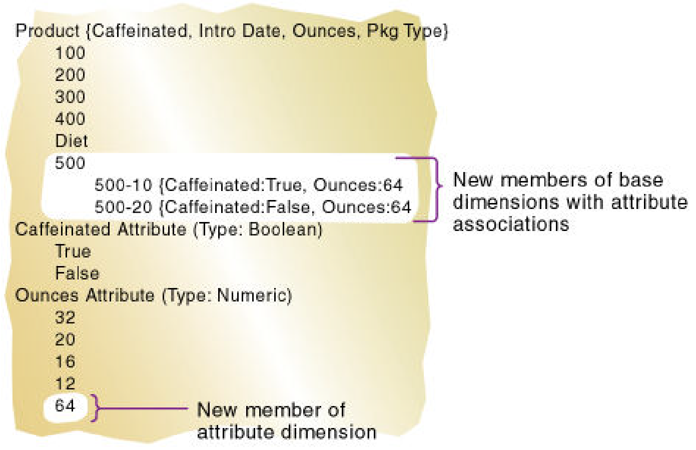
You can have Essbase use the attribute columns to build the members of the attribute dimensions. In Data Prep Editor, in the Dimension Build Settings tab of the Dimension Build Settings dialog box, clear the "Do not create members" option for the base dimension. See "Setting Member Properties" in the *Oracle Essbase Administration Services Online Help*.

When you are working with numeric ranges, you may need to build attribute dimensions and perform associations in separate steps. See "Working with Numeric Ranges" on page 325.

The Caffeinated3,Product field in Figure 79 shows how to associate attributes from additional single-level attribute dimensions. Because the base dimension is already specified, you need only to define an additional field for each attribute that you want to associate with the member of the base dimension.

The file in Figure 79 associates attributes as shown in the outline in Figure 80. The members 500, 500-10, and 500-20 are new members of the base dimension, Product. The member 64 is a new member of the Ounces attribute dimension.

Figure 80 Associating Attributes



```
Product {Caffeinated, Intro Date, Ounces, Pkg Type}
    100
    200
    300
    400
    Diet
    500
        500-10 {Caffeinated:True, Ounces:64
        500-20 {Caffeinated:False, Ounces:64
Caffeinated Attribute (Type: Boolean)
    True
    False
Ounces Attribute (Type: Numeric)
    32
    20
    16
    12
    64
```

New members of base dimensions with attribute associations
New member of attribute dimension

Updating Attribute Associations

You can also use the rules file shown in Figure 79 on page 321 to change attribute associations. Ensure that you allow association changes. In Data Prep Editor, on the Dimension Build Settings tab of the Dimension Build Settings dialog box, check “Allow association changes” for the base dimension. See “Setting Member Properties” in the *Oracle Essbase Administration Services Online Help*.

Note:

For duplicate member situations, the field to which the attribute is associated must contain the qualified member name. See “Building Qualified Member Names Through the Rules File” on page 340.

Removing Attribute Associations

To remove attribute associations, use the same process as for updating them, plus the following steps:

- In the Dimension Build Properties tab of the Field Properties dialog box, select “Delete when the field is empty” for the attribute field. (This option is ignored if “Allow association changes” is not selected.)
- Leave the field empty or NULL in the data source.

Working with Multilevel Attribute Dimensions

Multilevel, numeric, Boolean, and date attribute dimensions can have duplicate level 0 members. For example, associated with a Product dimension, you can have a Size attribute dimension with two levels. Level 1 categorizes sizes by men or by women. The level 0 members (attributes) are the actual sizes. You can have a member named 8 under Women and member named 8 under Men.

When an attribute is part of a multilevel numeric, Boolean, or date attribute dimension, the source data must include columns for all generations or levels of the attribute dimension. In the rules file, you must make copies of all fields that comprise the levels of the attribute dimension. Define the first set of attribute fields to build the attribute dimension. Define the second set of attribute fields to associate the attributes with the appropriate base dimension members. To ensure association with the correct attribute, indicate the parent field for the attribute field by making a copy of the parent field and setting the copy of the parent field as the field type Attribute Parent.

The position of the fields in the rules file is important.

- Place the copied attribute dimension field or fields that define the association immediately to the right of the field for the members of the base dimension.
- For a multilevel attribute dimension, place the attribute parent field immediately to the left of the field that is the child of the attribute parent.

The following steps describe how to define the fields in the rules file to build a multilevel attribute dimension and associate its members with members of its base dimension. This example uses the level references build method.

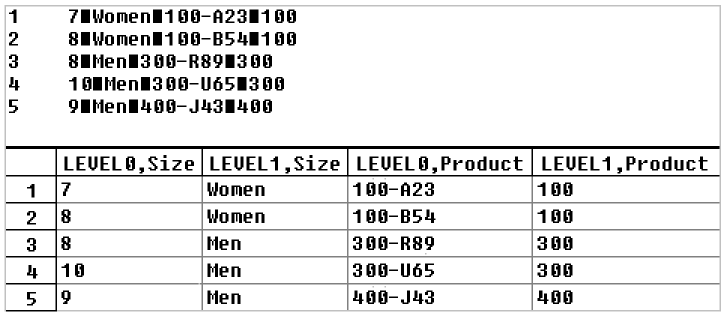
1. In the rules file, in field 1 and field 2, define the attribute dimension fields in the same way in which you define standard dimensions; specify type (level or generation), number, and dimension name.

   Essbase uses field1 and field2 to build the attribute dimension.

2. Define the fields for building the base dimension.

   In the following example, you are defining the level 0 and level 1 fields for the Product dimension. Figure 81 shows the fields of the rules file at this stage.

Figure 81 Defining Multilevel Attribute Dimensions Before Adding the Association Fields



```
1     7■Women■100-A23■100
2     8■Women■100-B54■100
3     8■Men■300-R89■300
4     10■Men■300-U65■300
5     9■Men■400-J43■400
```

	LEVEL0,Size	LEVEL1,Size	LEVEL0,Product	LEVEL1,Product
1	7	Women	100-A23	100
2	8	Women	100-B54	100
3	8	Men	300-R89	300
4	10	Men	300-U65	300
5	9	Men	400-J43	400

3. To define the association, make a copy of the field that contains the level 0 attribute.

   In the current example, make a copy of field 1.

   a. Use the attribute dimension name as the field type and specify the generation or level number of the member of the base dimension with which Essbase associates the attribute; for example, Size0.

   b. Specify the base dimension; for example, Product.

c. Move the new field immediately to the right of the field for the base dimension with which Essbase associates the attribute.

   In the current example, move the new field to the right of the field Level0, Product.

4. Make a copy of the field containing the parent of the attribute field.

   In the current example, make a copy of field 2.

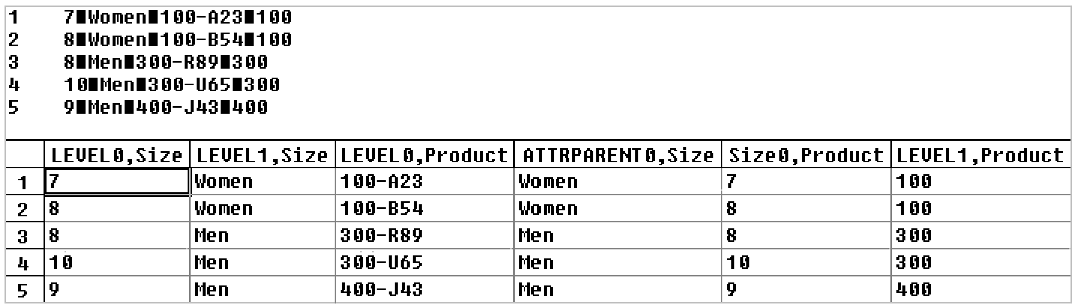
   a. Set the field type of the new field as Attribute Parent and specify the generation or level number of the base member with which you want Essbase to associate the attribute; for example, ATTRPARENT0.

   b. Specify the attribute dimension; for example, Size.

   c. Move the ATTRPARENT field immediately to the left of the attribute association field that you created in step 3.

As shown in Figure 82, the rules file now contains the field definitions to build the attribute dimension Size and to associate the members of Size with the appropriate members of the base dimension Product.

Figure 82 Source Data and Rules File for Building a Multilevel Attribute Dimension

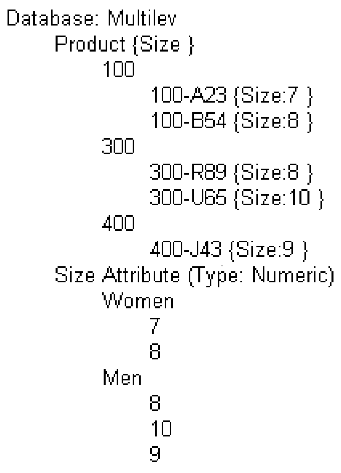


```
1    7■Women■100-A23■100
2    8■Women■100-B54■100
3    8■Men■300-R89■300
4    10■Men■300-U65■300
5    9■Men■400-J43■400
```

	LEVEL0,Size	LEVEL1,Size	LEVEL0,Product	ATTRPARENT0,Size	Size0,Product	LEVEL1,Product
1	7	Women	100-A23	Women	7	100
2	8	Women	100-B54	Women	8	100
3	8	Men	300-R89	Men	8	300
4	10	Men	300-U65	Men	10	300
5	9	Men	400-J43	Men	9	400

When you run a dimension build with the data shown in Figure 82, Essbase builds the Size attribute dimension and associates its members with the appropriate members of the base dimension. Figure 83 shows the updated outline.

Figure 83 Multilevel Attribute Dimension



```
Database: Multilev
    Product {Size }
        100
            100-A23 {Size:7 }
            100-B54 {Size:8 }
        300
            300-R89 {Size:8 }
            300-U65 {Size:10 }
        400
            400-J43 {Size:9 }
    Size Attribute (Type: Numeric)
        Women
            7
            8
        Men
            8
            10
            9
```

Working with Numeric Ranges

In many cases, you can use one rules file in a dimension build operation to dynamically build attribute dimensions for numeric ranges and to associate the members of the base dimension with the ranges. In the following situations, however, you must use two rules files: one to build the attribute dimension and one to associate the attributes with the appropriate members of the base dimension:

- When the range size is different for different members.

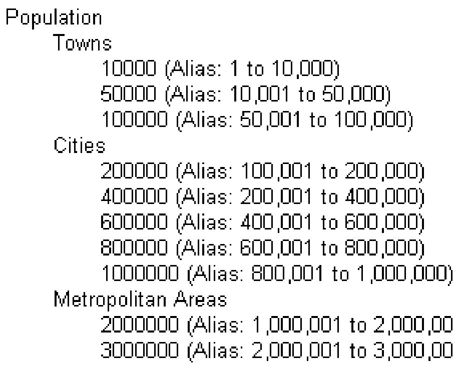
  For example, you can define small ranges for towns and cities with smaller populations, larger ranges for mid-sized cities, and ranges greater than 1,000,000 for cities with large populations.

- When the ranges are members of a multilevel attribute dimension.

  For example, the Population attribute dimension can have level 1 members that categorize the population ranges as Towns, Cities, and Metropolitan Areas.

The Population attribute dimension shown in Figure 84 demonstrates both situations. Population is a multilevel, numeric attribute dimension with level 0 members representing ranges of different sizes.

Figure 84 Numeric Attribute Dimension with Different-Sized Ranges



```
Population
    Towns
        10000 (Alias: 1 to 10,000)
        50000 (Alias: 10,001 to 50,000)
        100000 (Alias: 50,001 to 100,000)
    Cities
        200000 (Alias: 100,001 to 200,000)
        400000 (Alias: 200,001 to 400,000)
        600000 (Alias: 400,001 to 600,000)
        800000 (Alias: 600,001 to 800,000)
        1000000 (Alias: 800,001 to 1,000,000)
    Metropolitan Areas
        2000000 (Alias: 1,000,001 to 2,000,00
        3000000 (Alias: 2,000,001 to 3,000,00
```

You must use one rules file to build the Population dimension and another rules file to associate the Population dimension members as attributes of members of the base dimension.

Building Attribute Dimensions that Accommodate Ranges

First, create a rules file that uses the generation, level, or parent-child build method to build the attribute dimension. In the rules file, specify the following information:

- The name of the attribute dimension and its associated base dimension.
- The fields for building the attribute dimension.

  See "Setting Field Type Information" on page 277.

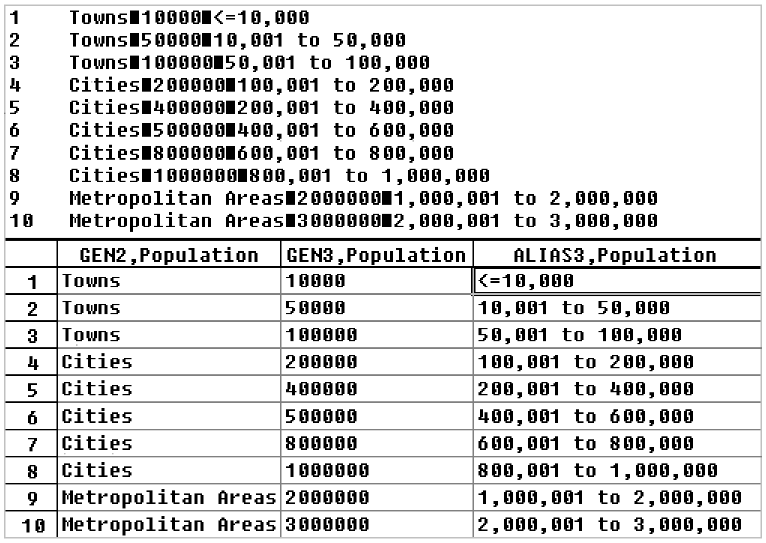
The source data must be in attribute sequence, in ascending order. If ranges have different sizes, the source data must include a record for every attribute range.

Note:

In later builds, you cannot insert attribute members between existing members.

To use the generation method to build the outline in Figure 84, you must sequence the source data in ascending sequence, based on the numeric attribute value. Define the fields in a rules file as shown in Figure 85. Additionally, Figure 85 shows how to associate aliases with attributes.

Figure 85 Rules File for Building a Numeric Attribute Dimension with Ranges



```
1    Towns■10000■<=10,000
2    Towns■50000■10,001 to 50,000
3    Towns■100000■50,001 to 100,000
4    Cities■200000■100,001 to 200,000
5    Cities■400000■200,001 to 400,000
6    Cities■500000■400,001 to 600,000
7    Cities■800000■600,001 to 800,000
8    Cities■1000000■800,001 to 1,000,000
9    Metropolitan Areas■2000000■1,000,001 to 2,000,000
10   Metropolitan Areas■3000000■2,000,001 to 3,000,000
```

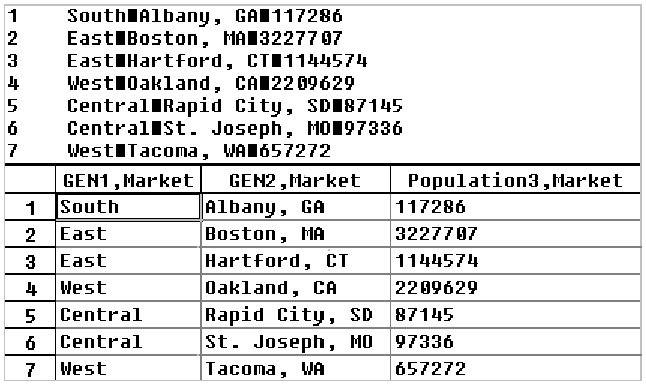
	GEN2,Population	GEN3,Population	ALIAS3,Population
1	Towns	10000	<=10,000
2	Towns	50000	10,001 to 50,000
3	Towns	100000	50,001 to 100,000
4	Cities	200000	100,001 to 200,000
5	Cities	400000	200,001 to 400,000
6	Cities	500000	400,001 to 600,000
7	Cities	800000	600,001 to 800,000
8	Cities	1000000	800,001 to 1,000,000
9	Metropolitan Areas	2000000	1,000,001 to 2,000,000
10	Metropolitan Areas	3000000	2,000,001 to 3,000,000

Associating Base Dimension Members with Their Range Attributes

After you build the numeric attribute dimension ranges, you need a rules file to associate the members of the base dimension with their attributes. The source data includes fields for the members of the base dimension and fields for the data values that Essbase uses to associate the appropriate Population attribute.

Define the rules file as shown in Figure 86.

Figure 86 Rules File for Associating Numeric Range Attributes



```
1    South■Albany, GA■117286
2    East■Boston, MA■3227707
3    East■Hartford, CT■1144574
4    West■Oakland, CA■2209629
5    Central■Rapid City, SD■87145
6    Central■St. Joseph, MO■97336
7    West■Tacoma, WA■657272
```

	GEN1,Market	GEN2,Market	Population3,Market
1	South	Albany, GA	117286
2	East	Boston, MA	3227707
3	East	Hartford, CT	1144574
4	West	Oakland, CA	2209629
5	Central	Rapid City, SD	87145
6	Central	St. Joseph, MO	97336
7	West	Tacoma, WA	657272

When you define the association field (for example, Population3, Market), place the attribute members within a range. In Data Prep Editor, on the Dimension Build Properties tab of the Field Properties dialog box, select "Place attribute members within a numeric range."

Note:

Figure 86 includes a city, Boston, whose population of 3,227,707 is outside the ranges of the attribute dimension in Figure 84 on page 325, where the ranges extend only to 3,000,000. To allow for values in the source data that are outside the ranges in the attribute dimension, enter a range size, such as 1000000. Essbase uses the range size to add members to the attribute dimension above the existing highest member or below the existing lowest member, as needed.

Caution!

After you associate members of the base dimension with members of the attribute dimension, if you manually insert new members into the attribute dimension or rename members of the attribute dimension, you may invalidate existing attribute associations. Consider an example where numeric range attributes are defined as "Tops of ranges" and an attribute dimension contains members 100, 200, 500, and 1000. A base dimension member with the value 556 is associated with the attribute 1000. If you rename a attribute dimension member from 500 to 600, the base dimension member with the value 556 now has an invalid association. This base member is still associated with the attribute 1000 when it should be associated with the attribute 600. If you manually insert new members or rename existing members, to ensure that associations are correct, rerun the dimension build procedure and associate the base members with the changed attribute dimensions. For example, rerunning the attribute association procedure correctly associates the member of the base dimension with the value 556 with the new attribute 600.

Ensuring the Validity of Associations

To ensure the validity of attribute associations, you must select the correct dimension building options and perform the builds in the proper sequence.

- **Adding or Changing Members of the Attribute Dimension:** After you associate members of a base dimension with their numeric attribute ranges, if you manually insert new members or rename existing members in the attribute dimension, ensure that associations between attributes and base members are correct by performing one of the following tasks:
  - Rerun the dimension build procedure that associates the base members with the changed attribute dimension.
  - Use Outline Editor to manually review and fix, as needed, the associations of all base dimensions.
- **Deleting Members from the Attribute Dimension:** You can delete all members of an attribute dimension so that you can rebuild the dimension with new data. In Data Prep Editor, on the Dimension Building Properties tab on the Field Properties dialog box, click the Ranges button and select "Delete all members of this attribute dimension." Essbase uses the start value and range size value to rebuild the attribute dimension. To ensure proper

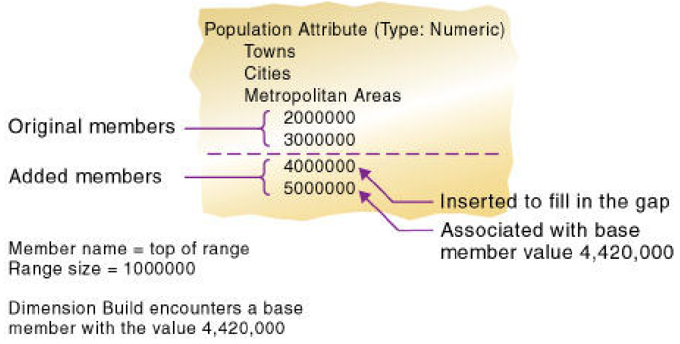
attribute association, on the Dimension Build Settings tab of the Dimension Build Settings dialog box, you must select the "Allow association changes" option for the base dimension.

- **Adding Members to the Base Dimension:** You can use the same rules file to add new members to the base dimension and to associate the new members with their numeric range attributes simultaneously. Provide a value for the range size. In Data Prep Editor, on the Dimension Build Properties tab in the Field Properties dialog box, click the Ranges button and specify the range size for the attribute dimension.

  If Essbase encounters a base dimension value that is greater than the highest attribute member by more than the range size or is lower than the lowest attribute member by more than the range size, it creates members in the attribute dimension to accommodate the out-of-range values.

  For example, in Figure 87, the numeric range attributes are defined as "Tops of ranges." The highest value member of the Population attribute dimension is 3000000. If the source data includes a record with the population 4,420,000, and the range size is 1000000, Essbase adds two members to the attribute dimension, 4000000 and 5000000, and associates the base member with the 5000000 attribute.

Figure 87 Dynamically Adding Attribute Range Members


Population Attribute (Type: Numeric)
Towns
Cities
Metropolitan Areas
2000000
3000000
4000000
5000000
Original members
Added members
Inserted to fill in the gap
Associated with base member value 4,420,000
Member name = top of range
Range size = 1000000
Dimension Build encounters a base member with the value 4,420,000

When you add range members and base dimension members at the same time, Essbase does not create aliases for the new members of the attribute dimension. If you want aliases that describe the range values for the new members of the attribute dimension, you must add the aliases in a separate operation.

Reviewing the Rules for Building Attribute and Base Dimensions

The information in this section describes areas unique to defining and associating attributes through a dimension build.

Getting Ready

- Before running a dimension build, you must define the attribute member name formats for the outline.

  See "Setting Member Names in Attribute Dimensions" on page 170.

- Defining new attribute dimensions in a rules file is different from defining new standard dimensions in a rules file.

Defining Fields in Rules Files

Rules files that are used to build single-level attribute dimensions require fewer field types than rules files that build and associate members of multilevel attribute dimensions.

- For single-level attribute dimensions, define the field that contains the attribute values as the field to be associated with the members of the base dimension. A dimension build uses the defined field to add new members to the attribute dimension.

  See "Associating Attributes" on page 321.

- For multilevel attribute dimensions, Essbase requires fields that define each generation or level in the attribute dimension and fields that define the associations. Use the new field type, Attribute Parent, to identify fields that are parent members for the attribute members being associated.

  See "Working with Multilevel Attribute Dimensions" on page 322.

Controlling Adding New Attribute Members

When Essbase encounters attribute data values that are not members of the attribute dimension, it automatically adds the values as new members. To prevent adding new members to attribute dimensions, in the Dimension Build Settings tab of the Dimension Build Settings dialog box, select the "Do not create members" option for the attribute dimension.

See "Setting Member Properties" in the *Oracle Essbase Administration Services Online Help*.

Controlling Associations

You can control the following associations:

- Making changes to attribute associations

  In Data Prep Editor, in the Dimension Build Settings tab of the Dimension Build Settings dialog box, select the "Allow association changes" option for the attribute dimension.

  See "Setting Member Properties" in the *Oracle Essbase Administration Services Online Help*.

- Enabling automatic association of base members with attributes that represent ranges of values

  In Data Prep Editor, on the Dimension Building Properties tab of the Field Properties dialog box, click the Ranges button and define the size of the range.

  See "Setting Field Type Information" on page 277.

- Concurrent attribute associations

  Use any build method except the "Add as build" methods.

  See "Understanding Build Methods" on page 309.

Note:

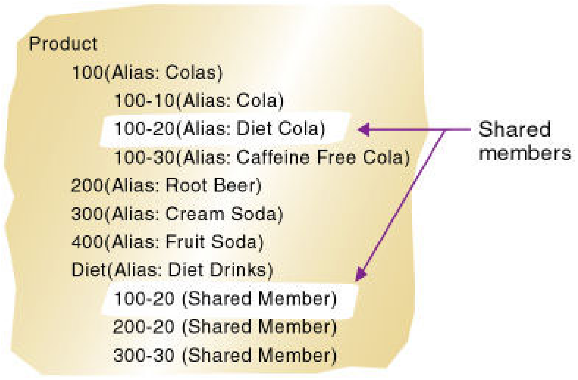
Because attributes are defined only in the outline, the data load process does not affect them.

Building Shared Members by Using a Rules File

The data associated with a shared member comes from an actual member with the same name as the shared member. Because the shared member stores a pointer to data contained in the actual member, the data is shared between the members and is stored only once.

For example, in the Sample.Basic database, the 100-20 (Diet Cola) member rolls up into the 100 family and into the Diet family.

Figure 88 Shared Members in the Sample.Basic Database



You can share members among as many parents as you want. Diet Cola has two parents (100 and Diet), but you can define it to roll up into more parents.

You can share members at multiple generations in the outline. In Figure 88, Diet Cola is shared by two members at generation 2 in the outline, but it can be shared by a member at generation 3 and a member at generation 4 as shown in Figure 96 on page 334.

Creating shared members at different generations in the outline is easy in Outline Editor; creating shared members using dimension build is more difficult. You must pick the build method and format the data source carefully.

The following sections describe how to build shared members in the outline by using a data source and a rules file.

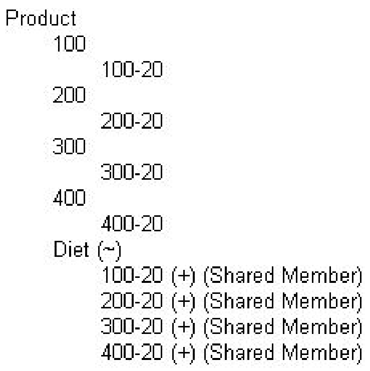
Note:

You should not create an outline in which a shared member is located before the actual member with which it is associated.

Sharing Members at the Same Generation

Members that are shared at the same generation roll up into the same branch. In the Sample.Basic database, 100-20 (Diet Cola) is shared by two parents (100 and Diet). Both parents roll up into the same branch (the Product dimension), and both parents are at generation 2, as shown in the following figure:

Figure 89 Members Shared at the Same Generation


```
Product
    100
        100-20
    200
        200-20
    300
        300-20
    400
        400-20
    Diet (~)
        100-20 (+) (Shared Member)
        200-20 (+) (Shared Member)
        300-20 (+) (Shared Member)
        400-20 (+) (Shared Member)
```

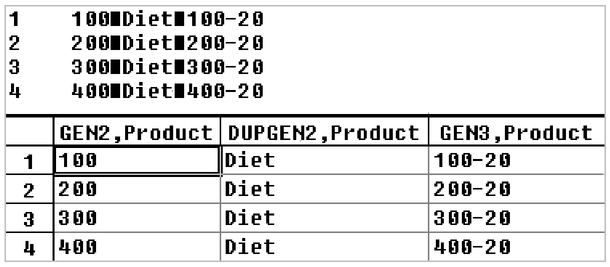
This scenario is the simplest way to share members. You can share members at the same generation by using any of the build methods discussed in the following sections.

Using Generation References to Create Same Generation Shared Members

To create shared member parents at the same generation by using the generation references build method, define the field type for the parent of the shared members as DUPGEN. A *duplicate generation* is a generation with shared members for children. Use the same GEN number as the primary member.

For example, to create the Diet parent and share the 100-20, 200-20, 300-20, and 400-20 members, use the sample data source file (`SHGENREF.TXT`) and set up the rules file so that the fields look like the sample rules file (`SHGENREF.RUL`) shown in Figure 90. 100 is the Cola family, 200 is the Root Beer family, 300 is the Cream Soda family, and the -20 after the family name indicates a diet version of the soda.

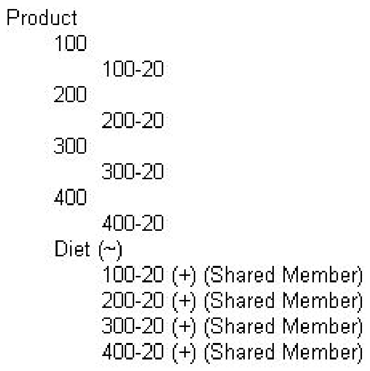
Figure 90 Sample Generation Shared Member Rules File


```
100■Diet■100-20
200■Diet■200-20
300■Diet■300-20
400■Diet■400-20
```

	GEN2,Product	DUPGEN2,Product	GEN3,Product
1	100	Diet	100-20
2	200	Diet	200-20
3	300	Diet	300-20
4	400	Diet	400-20

The data source and rules file illustrated in Figure 90 build the following tree:

Figure 91 Sample Generation Shared Member Rules Tree



```
Product
    100
        100-20
    200
        200-20
    300
        300-20
    400
        400-20
    Diet (~)
        100-20 (+) (Shared Member)
        200-20 (+) (Shared Member)
        300-20 (+) (Shared Member)
        400-20 (+) (Shared Member)
```

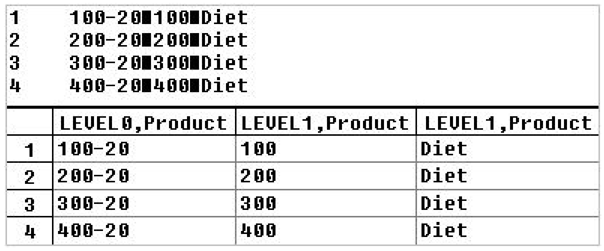
Using Level References to Create Same Generation Shared Members

To create shared members of the same generation by using the level references build method, first ensure that the primary and any secondary roll-ups are specified in one record. You can specify as many secondary roll-ups as you want, as long as they are all in one record.

Define the field type for the shared member as LEVEL. Then enter the level number. To create a shared member of the same generation, set the level number of the secondary roll-up to have the same number of levels as the primary roll-up. While processing the data source, Essbase creates a parent at the specified level and inserts the shared members under it.

For example, to create the shared 100-20 (Diet Cola), 200-20 (Diet Root Beer), 300-20 (Diet Cream Soda), and 400-20 (Diet Fruit Soda) members in the Sample.Basic database, use the sample data source file (`SHLEV.TXT`) and set up the rules file so that the fields look like the sample rules file (`SHLEV.RUL`) shown in Figure 92.

Figure 92 Sample Level Shared Member Rules File

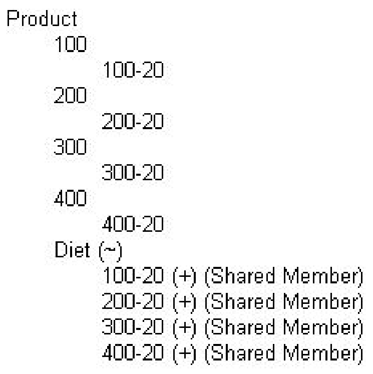


```
1    100-20■100■Diet
2    200-20■200■Diet
3    300-20■300■Diet
4    400-20■400■Diet
```

	LEVEL0,Product	LEVEL1,Product	LEVEL1,Product
1	100-20	100	Diet
2	200-20	200	Diet
3	300-20	300	Diet
4	400-20	400	Diet

The data source and rules file illustrated in Figure 92 build the following tree:

Figure 93 Sample Level Shared Member Rules Tree

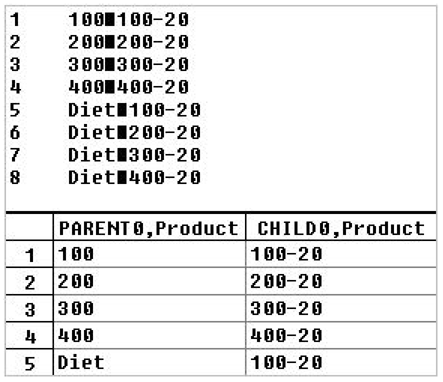


```
Product
    100
        100-20
    200
        200-20
    300
        300-20
    400
        400-20
    Diet (~)
        100-20 (+) (Shared Member)
        200-20 (+) (Shared Member)
        300-20 (+) (Shared Member)
        400-20 (+) (Shared Member)
```

Using Parent-Child References to Create Same Generation Shared Members

To create shared members of the same generation by using the parent-child references build method, define the PARENT and CHILD field types. Ensure that Essbase is set up to allow sharing (in the Dimension Build Settings tab of the Dimension Build Settings dialog box, clear the "Do not share" option). When sharing is enabled, Essbase automatically creates duplicate members under a new parent as shared members.

Figure 94 Sample Parent-Child Shared Members Rules File

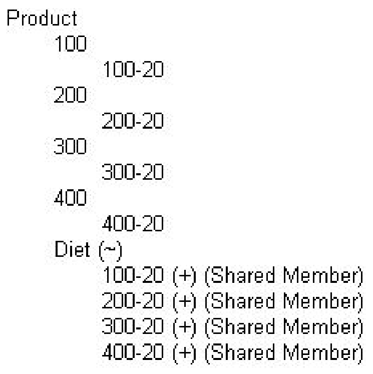


```
100■100-20
200■200-20
300■300-20
400■400-20
Diet■100-20
Diet■200-20
Diet■300-20
Diet■400-20
```

	PARENT0,Product	CHILD0,Product
1	100	100-20
2	200	200-20
3	300	300-20
4	400	400-20
5	Diet	100-20

The data source and rules file illustrated in Figure 94 build the following tree:

Figure 95 Sample Parent-Child Shared Member Rules Tree

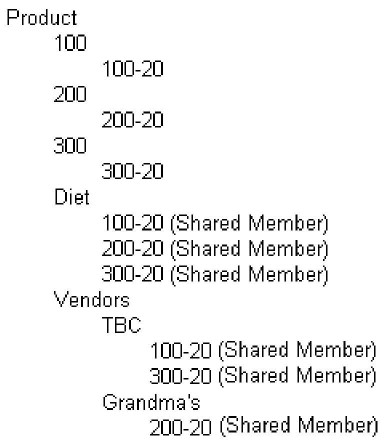


```
Product
    100
        100-20
    200
        200-20
    300
        300-20
    400
        400-20
    Diet (~)
        100-20 (+) (Shared Member)
        200-20 (+) (Shared Member)
        300-20 (+) (Shared Member)
        400-20 (+) (Shared Member)
```

Sharing Members at Different Generations

Sometimes you want shared members to roll up into parents that are at different generations in the outline. For example, in Figure 96, the shared members roll up into parents at generation 2 (Diet) and at generation 3 (TBC and Grandma's). This outline assumes that TBC (The Beverage Company) buys some beverages from outside vendors: it buys 200-20 (Diet Root Beer) from a vendor named Grandma's.

Figure 96 Members Shared at Different Generations



```
Product
    100
        100-20
    200
        200-20
    300
        300-20
    Diet
        100-20 (Shared Member)
        200-20 (Shared Member)
        300-20 (Shared Member)
    Vendors
        TBC
            100-20 (Shared Member)
            300-20 (Shared Member)
        Grandma's
            200-20 (Shared Member)
```

To share members across parents at different generations in the outline, use one of the following build methods.

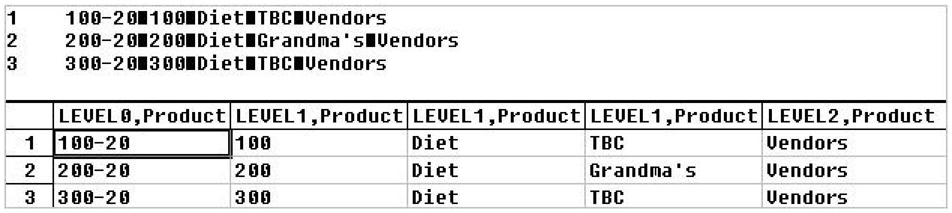
Using Level References to Create Different Generation Shared Members

To create shared members of different generations by using the level references build method, ensure that primary and secondary roll-ups are specified in one record. You can specify as many secondary roll-ups as you want, as long as they are all in one record.

Define the field type for the shared member as LEVEL. Then enter the level number. While processing the data source, Essbase creates a parent at the specified level and inserts the shared members under it.

For example, to share the products 100-20, 200-20, and 300-20 with a parent called Diet and two parents called TBC and Grandma's, use the sample data file and the rules file shown in Figure 97:

Figure 97 Level References Sample Rules File for Shared Members at Different Generations



```
1     100-20■100■Diet■TBC■Vendors
2     200-20■200■Diet■Grandma's■Vendors
3     300-20■300■Diet■TBC■Vendors
```

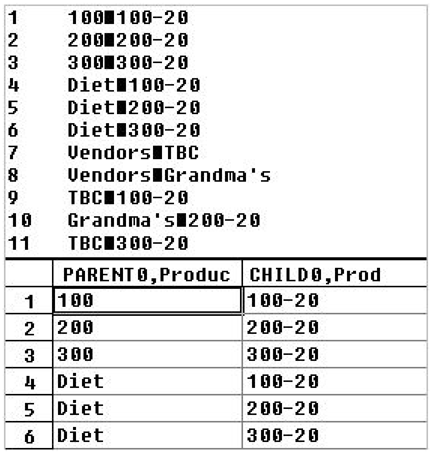
	LEVEL0,Product	LEVEL1,Product	LEVEL1,Product	LEVEL1,Product	LEVEL2,Product
1	100-20	100	Diet	TBC	Vendors
2	200-20	200	Diet	Grandma's	Vendors
3	300-20	300	Diet	TBC	Vendors

The data source and rules file illustrated in Figure 97 build the tree illustrated in Figure 96.

Using Parent-Child References to Create Different Generation Shared Members

To create shared members at the different generation using the parent-child references build method, define the PARENT and CHILD field types. Ensure that Essbase is set up to allow sharing (on the Dimension Build Settings tab of the Dimension Build Settings dialog box, clear the “Do not share” option). When sharing is enabled, Essbase automatically creates duplicate members under a new parent as shared members.

Figure 98 Parent-Child References Sample Rules File for Shared Members at Different Generations



```
1     100■100-20
2     200■200-20
3     300■300-20
4     Diet■100-20
5     Diet■200-20
6     Diet■300-20
7     Vendors■TBC
8     Vendors■Grandma's
9     TBC■100-20
10    Grandma's■200-20
11    TBC■300-20
```

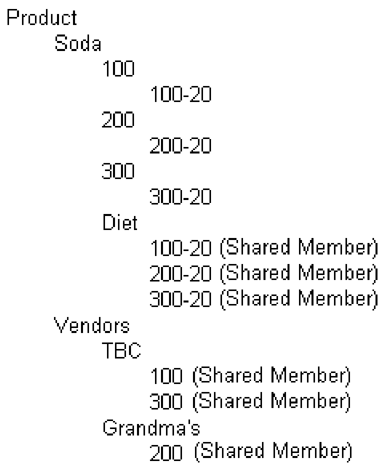
	PARENT0,Produc	CHILD0,Prod
1	100	100-20
2	200	200-20
3	300	300-20
4	Diet	100-20
5	Diet	200-20
6	Diet	300-20

The data source and rules file illustrated in Figure 98 build the tree illustrated in Figure 96.

Sharing Non-Level 0 Members

Sometimes you want to share non-level 0 members (members that are not at the lowest generation). For example, in Figure 99, 100, 200, and 300 are shared by TBC and Grandma’s. This outline assumes that TBC buys some of its product lines from outside vendors: it buys 200 (all root beer) from a vendor named Grandma’s.

Figure 99 Non-Level 0 Members Shared at Different Generations



```
Product
    Soda
        100
            100-20
        200
            200-20
        300
            300-20
        Diet
            100-20 (Shared Member)
            200-20 (Shared Member)
            300-20 (Shared Member)
    Vendors
        TBC
            100 (Shared Member)
            300 (Shared Member)
        Grandma's
            200 (Shared Member)
```

To share non-level 0 members, use one of the following build methods:

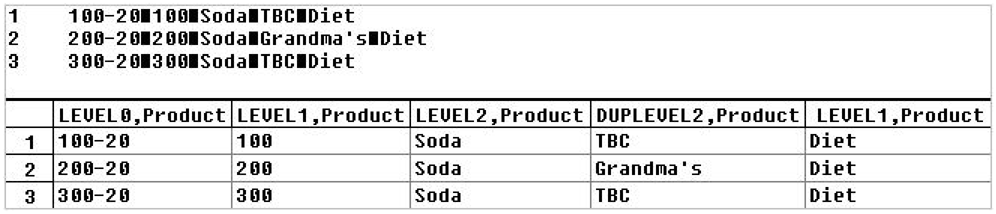
Using Level References to Create Non-Level 0 Shared Members

To create shared non-level 0 members by using the level references build method, ensure that primary and secondary roll-ups are specified in one record. You can specify unlimited secondary roll-ups, as long as they are all in one record.

Define the field type for the parent of the shared member as duplicate level (DUPLEVEL), and then enter the level number. To create a shared member of the same generation, set the level number of the secondary roll-up to have the same number of levels as the primary roll-up. While processing the data source, Essbase creates a parent at the specified level and inserts the shared members under it.

For example, to share the product lines 100, 200, and 300 with a parent called Soda and parents called TBC and Grandma's, use the sample data file and rules file shown in Figure 100. This data source and rules file work only if the Diet, TBC, and Grandma's members exist in the outline. The DUPLEVEL field is always created as a child of the dimension (at generation 2), unless the named level field already exists in the outline.

Figure 100 Level References Sample Rules File for Non-Level 0 Shared Members at Different Generations



```
100-20■100■Soda■TBC■Diet
200-20■200■Soda■Grandma's■Diet
300-20■300■Soda■TBC■Diet
```

	LEVEL0,Product	LEVEL1,Product	LEVEL2,Product	DUPLEVEL2,Product	LEVEL1,Product
1	100-20	100	Soda	TBC	Diet
2	200-20	200	Soda	Grandma's	Diet
3	300-20	300	Soda	TBC	Diet

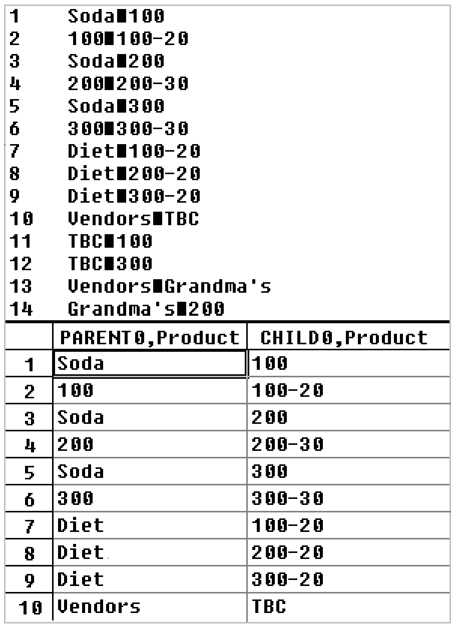
The data source and rules file illustrated in Figure 100 build the tree illustrated in Figure 99.

Using Parent-Child References to Create Non-Level 0 Shared Members

The parent-child references build method is the most versatile for creating shared members. It does not have any restrictions on the position of the shared members in the outline, unlike the generation references and level references build methods.

To create shared non-level 0 members at the same generation using the parent-child references build method, define the PARENT and CHILD field types. Ensure that Essbase is set up to allow sharing (clear “Do Not Share” in the Dimension Build Settings tab of the Dimension Build Settings dialog box). When sharing is enabled, Essbase automatically creates duplicate members under a new parent as shared members.

Figure 101 Parent-Child Sample Rules File for Non-Level 0 Shared Members



```
Soda■100
100■100-20
Soda■200
200■200-30
Soda■300
300■300-30
Diet■100-20
Diet■200-20
Diet■300-20
Vendors■TBC
TBC■100
TBC■300
Vendors■Grandma's
Grandma's■200
```

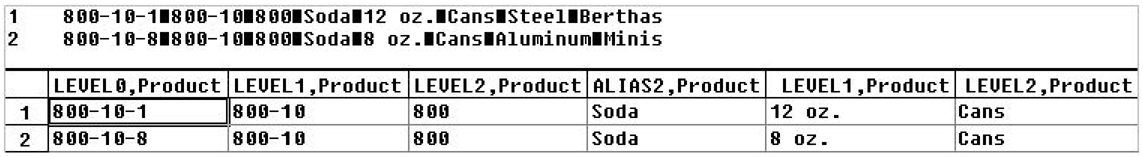
	PARENT0,Product	CHILD0,Product
1	Soda	100
2	100	100-20
3	Soda	200
4	200	200-30
5	Soda	300
6	300	300-30
7	Diet	100-20
8	Diet	200-20
9	Diet	300-20
10	Vendors	TBC

The data source and rules file illustrated in Figure 101 build the tree illustrated in Figure 99.

Building Multiple Roll-Ups by Using Level References

To enable the retrieval of totals from multiple perspectives, you can also put shared members at different levels in the outline. Use the level references build method. For example, the rules file (`LEVELMUL.RUL`) shown in Figure 102 specifies build instructions for levels in the Product dimension:

Figure 102 Rules File Fields Set to Build Multiple Roll-Ups Using Level References

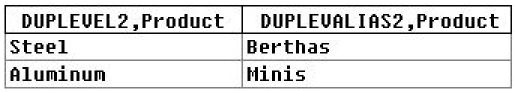


```
800-10-1■800-10■800■Soda■12 oz.■Cans■Steel■Berthas
800-10-8■800-10■800■Soda■8 oz.■Cans■Aluminum■Minis
```

	LEVEL0,Product	LEVEL1,Product	LEVEL2,Product	ALIAS2,Product	LEVEL1,Product	LEVEL2,Product
1	800-10-1	800-10	800	Soda	12 oz.	Cans
2	800-10-8	800-10	800	Soda	8 oz.	Cans

Because the record is so long, this second graphic shows the rules file scrolled to the right to show the extra members:

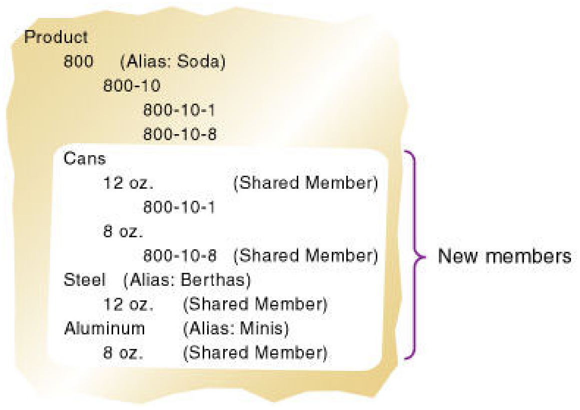
Figure 103 Scrolled Window



DUPLEVEL2,Product	DUPLEVALIAS2,Product
Steel	Berthas
Aluminum	Minis

When you run the dimension build using the data in Figure 103, Essbase builds the following member tree:

Figure 104 Multiple Roll-Ups



```
Product
    800    (Alias: Soda)
        800-10
            800-10-1
            800-10-8
    Cans
        12 oz.            (Shared Member)
            800-10-1
        8 oz.
            800-10-8  (Shared Member)
    Steel   (Alias: Berthas)
        12 oz.     (Shared Member)
    Aluminum       (Alias: Minis)
        8 oz.      (Shared Member)
```

This example enables analysis not only by package type (Cans), but also by packaging material (comparing sales of aluminum cans and steel cans).

Because Product is a sparse dimension, you can use an alternative outline design to enable retrieval of the same information. For example, consider creating a multilevel attribute dimension for package type with Steel and Aluminum as level 0 members under Can. For outline design guidelines, see "Analyzing Database Design" on page 86.

Creating Shared Roll-Ups from Multiple Data Sources

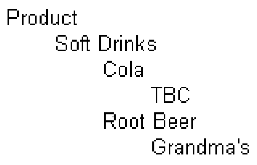
In many situations, the data for a dimension is in multiple data sources. If you are building dimensions from more than one data source and want to create multiple roll-ups, load the first data source using the most appropriate build method, and then load all other data sources using the parent-child references build method. Ensure that Essbase is set up to allow sharing (clear "Do Not Share" on the Dimension Build Settings tab of the Dimension Build Settings dialog box).

For example, using the following Product data source:

```
"Soft Drinks"   Cola
"Soft Drinks"   "Root Beer"
Cola            TBC
"Root Beer"     Grandma's
```

Essbase builds the tree illustrated in Figure 105:

Figure 105 Soft Drinks Tree



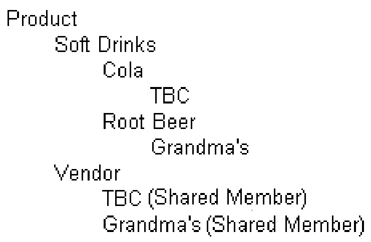
```
Product
    Soft Drinks
        Cola
            TBC
        Root Beer
            Grandma's
```

Then load the second data source below to relate the products to the vendors using the parent-child build method. Ensure that Essbase is set up to allow sharing.

```
Vendor   TBC
Vendor   Grandma's
```

Essbase builds the tree illustrated in Figure 106:

Figure 106 Shared Roll-Ups Tree



```
Product
    Soft Drinks
        Cola
            TBC
        Root Beer
            Grandma's
    Vendor
        TBC (Shared Member)
        Grandma's (Shared Member)
```

Building Duplicate Member Outlines

Duplicate member outlines contain multiple members with the same name, where the values are not shared. In unique member outlines, only shared members can have the same name. See Chapter 8, "Creating and Working With Duplicate Member Outlines."

The rules file enables you to set whether dimensions, levels, and generations in a duplicate member outline are unique or can include duplicate members.

➤ To set dimension uniqueness during a dimension build, see "Setting Dimension Properties" in the *Oracle Essbase Administration Services Online Help*.

In the rules file, generation or level uniqueness is set on the Generation/Level Names tab of the Dimension Properties dialog box.

➤ To set generation or level uniqueness during a dimension build, see "Setting Dimension Properties" in the *Oracle Essbase Administration Services Online Help*.

Uniquely Identifying Members Through the Rules File

To ensure that duplicate member outline hierarchies are built correctly, use qualified member names in the data source or use the rules file to construct qualified member names from fields in the data source.

In most situations, the reference method and arrangement of fields provide enough information for Essbase to map data source columns to members in a duplicate member outline. The dimension build rules file for duplicate member is similar to the rules file for unique member outlines.

The following operations require more complex rules files for qualified member names:

- **Parent-child dimension builds:** The parent-child build method requires two fields, one for parent and one for child. For duplicate member situations the parent field must contain the qualified member name. Parent-child dimension builds on duplicate member outlines do not support attribute association. To ensure that attributes are associated with the correct members, perform the associations in a separate pass, using a generation-reference or level-reference rules file.
- **Association of attributes to existing members in the outline:** For duplicate member situations, the field to which the attribute is associated must contain the qualified member name.

Building Qualified Member Names Through the Rules File

If the data source does not contain the qualified member name as a field, you can use the rules file to edit and join multiple fields resulting in qualified member names.

To create qualified member names, use the Field Edits tab of the Data Source Properties dialog box to copy, move, and join fields, and to create brackets and periods. For example, you can assign population attributes to existing city members in a duplicate member dimension. You can use move, join, and create operations to build the qualified name.

For example, the cities in the following data source already exist in the outline. You want to associate with the cities the population attributes in the last column of this four-column data source:

```
Central "Kansas City" Kansas     706010
Central "Kansas City" Missouri   1070052
East    "New York"    "New York" 8104079
```

Editing this source through the rules file to build qualified names and sequence the field columns properly involves the following edits:

- Using Create using text operations, create one field each for the following text elements:
  - [

    - `].[`
    - `].[`
    - `]`

- Using Move operations, move the fields to the following sequence:

```
1   2         3     4         5    6            7   8
[   Central   ].[   Kansas   ].[  Kansas City  ]   706010
```

- Using Join operations, join together fields 1 2 3 4 5 6 7 to create the single field to which the attributes are to be associated: [Central].[Kansas].[Kansas City]. The rules file now shows two fields:

```
1                               2
 [Central].[Kansas].[Kansas City]  706010
```

Part IV

Calculating Data

In Calculating Data:

21 Calculating Essbase Databases

In This Chapter

The information in this chapter applies only to block storage databases and is not relevant to aggregate storage databases.

Also see:

- Chapter 57, "Comparison of Aggregate and Block Storage."
- "Calculating Aggregate Storage Databases" on page 971

About Database Calculation

A database contains two types of values:

- Values that you enter, called *input data*
- Values that are calculated from input data

For example:

- You enter regional sales figures for a variety of products. You calculate the total sales for each.
- You enter the budget and actual values for the cost of goods sold for several products in several regions. You calculate the variance between budget and actual values for each product in each region.
- The database contains regional sales figures and prices for all products. You calculate what happens to total profit if you increase the price of one product in one region by 5%.

Small differences in the precision of cell values may occur between calculations run on different platforms, due to operating system math library differences.

Note:

Most computers represent numbers in binary, and therefore can only approximately represent real numbers. Because binary computers cannot hold an infinite number of bits after a decimal point, numeric fractions such as one-third (0.3333...) cannot be expressed as a decimal with a terminating point. Fractions with a denominator of the power of two (for example, 0.50) or ten (0.10) are the only real numbers that can be represented exactly. See IEEE Standard 754 for Floating-Point Representation (IEEE, 1985).

Essbase offers two methods for calculating a database:

- Outline calculation
- Calculation script calculation

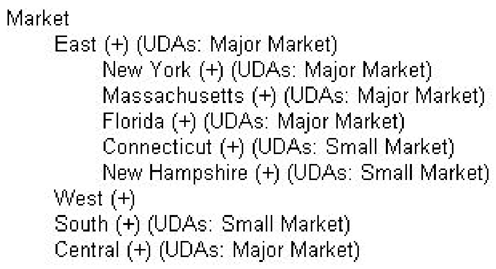
The method that you choose depends on the type of calculation that you want to perform.

Outline Calculation

Outline calculation is the simplest calculation method. Essbase bases the calculation of the database on the relationships between members in the database outline and on any formulas that are associated with members in the outline.

For example, Figure 107 shows the relationships between the members of the Market dimension in the Sample.Basic database. The values for New York, Massachusetts, Florida, Connecticut, and New Hampshire are added to calculate the value for East. The values for East, West, South, and Central are added to calculate the total value for Market.

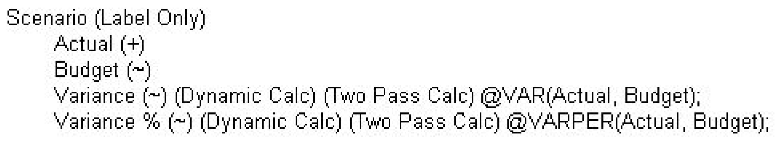
Figure 107 Relationship Between Members of the Market Dimension



```
Market
    East (+) (UDAs: Major Market)
        New York (+) (UDAs: Major Market)
        Massachusetts (+) (UDAs: Major Market)
        Florida (+) (UDAs: Major Market)
        Connecticut (+) (UDAs: Small Market)
        New Hampshire (+) (UDAs: Small Market)
    West (+)
    South (+) (UDAs: Small Market)
    Central (+) (UDAs: Major Market)
```

Figure 108 shows the Scenario dimension from the Sample.Basic database. The Variance and Variance % members are calculated by using the formulas attached to them.

Figure 108 Calculation of Variance and Variance %



```
Scenario (Label Only)
    Actual (+)
    Budget (~)
    Variance (~) (Dynamic Calc) (Two Pass Calc) @VAR(Actual, Budget);
    Variance % (~) (Dynamic Calc) (Two Pass Calc) @VARPER(Actual, Budget);
```

It may be more efficient to calculate some member combinations when you retrieve the data instead of calculating the member combinations during the regular database calculation. You can use dynamic calculations to calculate data at retrieval time. See Chapter 26, “Dynamically Calculating Data Values.”

Calculation Script Calculation

Calculation script calculation is the second method of calculation. Using a calculation script, you can choose exactly how to calculate a database. For example, you can calculate part of a database or copy data values between members.

A calculation script contains a series of calculation commands, equations, and formulas. For example, the following calculation script increases the actual marketing expenses in the New York region by 5%.

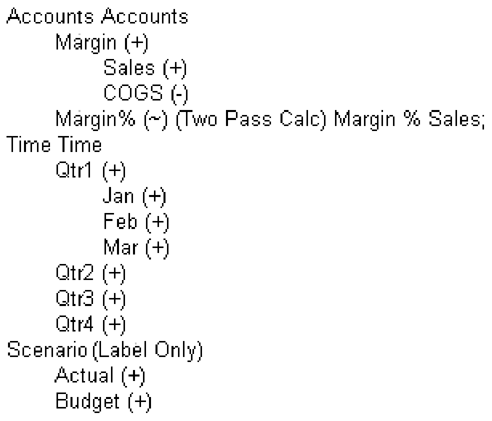
```
FIX (Actual, “New York”)
     Marketing = Marketing *1.05;
ENDFIX;
```

See Chapter 28, “Developing Calculation Scripts.”

About Multidimensional Calculation Concepts

Figure 109, which is based on a simplified database, illustrates the nature of multidimensional calculations:

Figure 109 Calculating a Multidimensional Database



```
Accounts Accounts
     Margin (+)
          Sales (+)
          COGS (-)
     Margin% (~) (Two Pass Calc) Margin % Sales;
Time Time
     Qtr1 (+)
          Jan (+)
          Feb (+)
          Mar (+)
     Qtr2 (+)
     Qtr3 (+)
     Qtr4 (+)
Scenario (Label Only)
     Actual (+)
     Budget (+)
```

The database has three dimensions—Accounts, Time, and Scenario.

The Accounts dimension has four members:

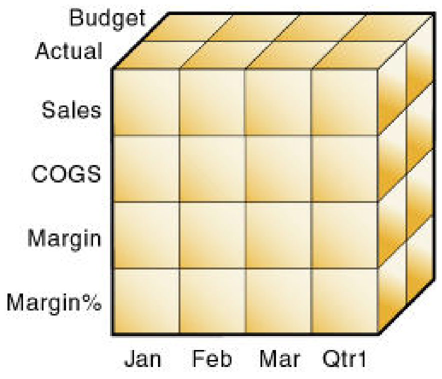
- Sales and COGS are input values
- Margin = Sales - COGS
- Margin% = Margin % Sales (Margin as a percentage of Sales)

The Time dimension has four quarters. The example displays only the members in Qtr1—Jan, Feb, and Mar.

The Scenario dimension has two child members—Budget for budget values and Actual for actual values.

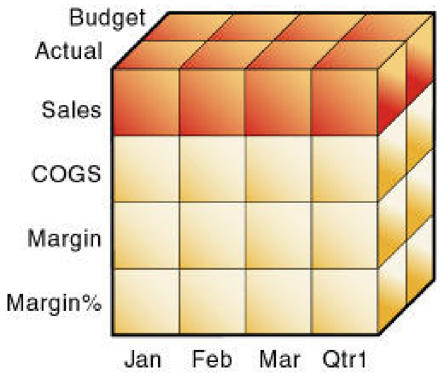
An intersection of members (one member on each dimension) represents a data value. The following example has three dimensions; therefore, the dimensions and data values in the database can be represented as a cube, as shown in Figure 110:

Figure 110 Three-Dimensional Database



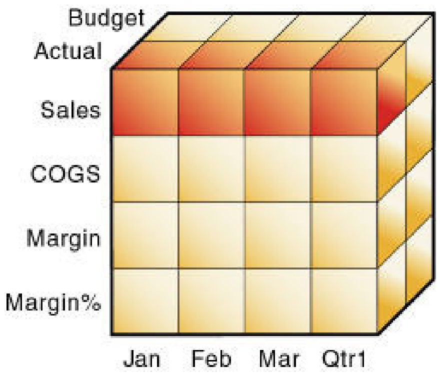
As shown in Figure 111, when you refer to Sales, you are referring to a slice of the database containing eight Sales values.

Figure 111 Sales, Actual, Budget Slice of the Database



As shown in Figure 112, when you refer to Actual Sales, you are referring to four Sales values:

Figure 112 Actual, Sales Slice of the Database

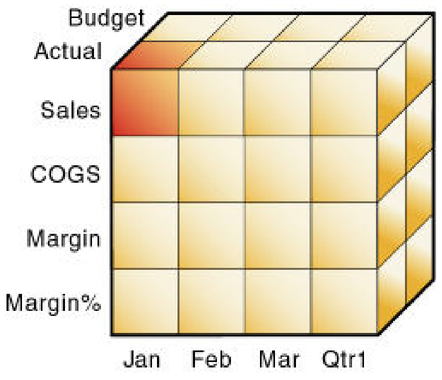


To refer to a specific data value in a multidimensional database, you must specify each member on each dimension. A data value is stored in one cell in the database. In Figure 113, the cell containing the data value for Sales, Jan, Actual is shaded.

In Essbase, member combinations are denoted by a cross-dimensional operator (->). Sales, Jan, Actual is written as:

```
Sales -> Jan -> Actual
```

Figure 113 Sales, Jan, Actual Slice of the Database



When Essbase calculates the formula “Margin% = Margin % Sales,” it takes each Margin value and calculates it as a percentage of its corresponding Sales value.

Essbase cycles through the database and calculates Margin% as follows:

1. Margin -> Jan -> Actual as a percentage of Sales -> Jan -> Actual. The result is placed in Margin% -> Jan -> Actual.
2. Margin -> Feb -> Actual as a percentage of Sales -> Feb -> Actual. The result is placed in Margin% -> Feb -> Actual.
3. Margin -> Mar -> Actual as a percentage of Sales -> Mar -> Actual. The result is placed in Margin% -> Mar -> Actual.
4. Margin -> Qtr1 -> Actual as a percentage of Sales -> Qtr1 -> Actual. The result is placed in Margin% -> Qtr1 -> Actual.

5. Margin -> Jan -> Budget as a percentage of Sales -> Jan -> Budget. The result is placed in Margin% -> Jan -> Budget.
6. Essbase continues cycling through the database until it has calculated Margin% for every combination of members in the database.

For information about how Essbase calculates a database, see Chapter 24, "Defining Calculation Order."

Setting the Default Calculation

By default, the calculation for a database is a CALC ALL of the database outline. CALC ALL consolidates all dimensions and members and calculates all formulas in the outline.

You can, however, specify any calculation script as the default database calculation. Thus, you can assign a frequently used script to the database rather than loading the script each time you want to perform its calculation. If you want a calculation script to work with calculation settings defined at the database level, you must set the calculation script as the default calculation.

➤ To set the default calculation, use a tool:

Tool	Topic	Location
Administration Services	Setting the Default Calculation	*Oracle Essbase Administration Services Online Help*
MaxL	**alter database**	*Oracle Essbase Technical Reference*
ESSCMD	SETDEFAULTCALCFILE	*Oracle Essbase Technical Reference*

Calculating Databases

If you have Calculation permissions, you can calculate a database.

When you use Administration Services to calculate a database, you can execute the calculation in the background so that you can continue working as the calculation processes. You can then check the status of the background process to see when the calculation is complete.

➤ To calculate a database, use a tool:

Tool	Topic	Location
Administration Services	Calculating Block Storage Databases	*Oracle Essbase Administration Services Online Help*
MaxL	**execute calculation**	*Oracle Essbase Technical Reference*
ESSCMD	CALC, CALCDEFAULT, and CALCLINE	*Oracle Essbase Technical Reference*
Spreadsheet Add-in	Calculating a Database	*Oracle Essbase Spreadsheet Add-in Online Help*

Canceling Calculations

➤ To stop a calculation before Essbase completes it, click the **Cancel** button while the calculation is running.

When you cancel a calculation, Essbase performs one of the following operations:

- Reverts all values to their previous state
- Retains any values calculated before the cancellation

How Essbase handles the cancellation depends on the Essbase Kernel Isolation Level settings. See "Understanding Isolation Levels" on page 779.

Parallel and Serial Calculation

Essbase supports parallel and serial calculations:

- *Serial calculation (default):* All steps in a calculation run on a single thread. Each task is completed before the next is started.
- *Parallel calculation:* The Essbase calculator can analyze a calculation, and, if appropriate, assign tasks to multiple CPUs (up to four).

See "Using Parallel Calculation" on page 866.

Security Considerations

To calculate a database, you must have Calculate permissions for the database outline. With calculate permissions, you can calculate any value in the database, and you can calculate a value even if a security filter denies you read and update permissions. Carefully consider providing users with calculate permissions.

See Chapter 38, "User Management and Security."

22

Developing Formulas

In This Chapter

The information in this chapter applies only to block storage databases and is not relevant to aggregate storage databases.

Also see:

- Chapter 57, "Comparison of Aggregate and Block Storage"
- "Developing Formulas on Aggregate Storage Outlines" on page 930
- Chapter 23, "Reviewing Examples of Formulas"
- Chapter 55, "Optimizing Calculations"

Understanding Formulas

Formulas calculate relationships between members in a database outline. With formulas, you can:

- Apply formulas to members in the database outline. Use this method if you do not need to control database calculations carefully for accuracy or performance. This method limits formula size to less than 64 KB.

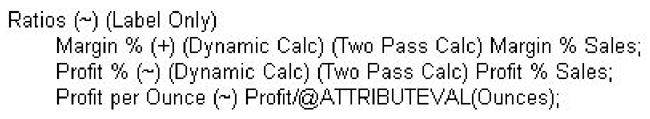
  See "Composing Formulas" on page 359.

- Place formulas in a calculation script. Use this method if you need to control database calculations carefully.

  See "Using Formulas in a Calculation Script" on page 452.

Figure 114 shows the Measures dimension from the Sample.Basic database. The Margin %, Profit %, and Profit per Ounce members are calculated using the formulas applied to them.

Figure 114 Calculation of Margin %, Profit %, and Profit per Ounce



```
Ratios (~) (Label Only)
    Margin % (+) (Dynamic Calc) (Two Pass Calc) Margin % Sales;
    Profit % (~) (Dynamic Calc) (Two Pass Calc) Profit % Sales;
    Profit per Ounce (~) Profit/@ATTRIBUTEVAL(Ounces);
```

Essbase provides a comprehensive set of operators and functions, which you can use to construct formula calculations on a database. The topics in this section describe the elements you can place in a formula and provide basic information about formula calculation and syntax:

Operators

Table 46 shows the types of operators you can use in formulas:

Table 46 Descriptions of Operator Types

Operator Type	Description
Mathematical	Perform common arithmetic operations. For example, you can add, subtract, multiply, or divide values. For a list of mathematical operators, see the *Oracle Essbase Technical Reference*.
Conditional	Control the flow of formula executions based on the results of conditional tests. For example, you can use an IF statement to test for a specified condition. For a list of conditional operators, see the *Oracle Essbase Technical Reference*. For information on writing conditional formulas, see "Conditional Tests" on page 360.
Cross-dimensional	Point to the data values of specific member combinations. For example, point to the sales value for a specific product in a specific region. See "Working with Member Combinations Across Dimensions" on page 371.

For information about using operators with #MISSING, zero, and other values, see the "Essbase Functions" section in the *Oracle Essbase Technical Reference*.

Functions

Functions are predefined routines that perform specialized calculations and return sets of members or data values. The following table shows the types of functions you can use in formulas.

Table 47 Descriptions of Function Types

Function Type	Description
Boolean	Provide a conditional test by returning a TRUE (1) or FALSE (0) value. For example, you can use the @ISMBR function to determine whether the current member matches any members specified.
Mathematical	Perform specialized mathematical calculations.

Function Type	Description
	For example, you can use the @AVG function to return the average value of a list of members.
Relationship	Look up data values within a database during a calculation. For example, you can use the @ANCESTVAL function to return the ancestor values of a specified member combination.
Range	Declare a range of members as an argument to another function or command. For example, you can use the @SUMRANGE function to return the sum of all members within a specified range.
Financial	Perform specialized financial calculations. For example, you can use the @INTEREST function to calculate simple interest or the @PTD function to calculate period-to-date values.
Member Set	Generate a list of members that is based on a specified member. For example, you can use the @ICHILDREN function to return a specified member and its children.
Allocation	Allocate values that are input at a parent level across child members. You can allocate values within the same dimension or across multiple dimensions. For example, you can use the @ALLOCATE function to allocate sales values that are input at a parent level to the children of the parent; the allocation of each child is determined by its share of the sales of the previous year.
Forecasting	Manipulate data for the purposes of smoothing or interpolating data, or calculating future values. For example, you can use the @TREND function to calculate future values that are based on curve-fitting to historical values.
Statistical	Calculate advanced statistics. For example, you can use the @RANK function to calculate the rank of a specified member or a specified value in a data set.
Date and Time	Use date and time characteristics in calculation formulas. For example, you can use the @TODATE function to convert date strings to numbers that can be used in calculation formulas.
Miscellaneous	This type provides two kinds of functionality: ● You can specify calculation modes that Essbase is to use to calculate a formula—cell, block, bottom-up, and top-down. ● You can manipulate character strings for member and dimension names; for example, to generate member names by adding a character prefix to a name or removing a suffix from a name, or by passing the name as a string.
Custom-defined Functions	This type enables you to perform functions that you develop for calculation operations. These custom-developed functions are written in the Java programming language and are called by the Essbase calculator framework as external functions.

For a complete list of operators, functions, and syntax, see the *Oracle Essbase Technical Reference*. Also see Chapter 23, "Reviewing Examples of Formulas."

Note:

Abbreviations of functions are not supported. Some commands may work in an abbreviated form, but if another function has a similar name, Essbase may use the wrong function. Use the complete function name to ensure correct results.

Dimension and Member Names

You can include dimension and member names in a formula, as illustrated in the following example:

```
Scenario
100-10
Feb
```

Constant Values

You can assign a constant value to a member:

```
California = 120;
```

In this formula, California is a member in a sparse dimension and 120 is a constant value. Essbase automatically creates all possible data blocks for California and assigns the value 120 to all data cells. Many thousands of data blocks may be created.

To assign constants in a sparse dimension to only those intersections that require a value, use a FIX statement. See "Constant Values Assigned to Members in a Sparse Dimension" on page 876.

Nonconstant Values

If you assign anything other than a constant to a member in a sparse dimension, and no data block exists for that member, new blocks are not created unless Essbase is enabled to create blocks on equations.

For example, to create blocks for West that did not exist before running the calculation, you must enable Create Blocks on Equations for this formula:

```
West = California + 120;
```

You can enable Create Blocks on Equations at the database level, whereby blocks are always created, or you can control block creation within calculation scripts.

➤ To enable the Create Blocks on Equations feature for all calculation scripts for a specific database, use a tool:

Tool	Topic	Location
Administration Services	Enabling Create Blocks on Equations	*Oracle Essbase Administration Services Online Help*

Tool	Topic	Location
MaxL	**alter database**	*Oracle Essbase Technical Reference*
ESSCMD	SETDBSTATE	*Oracle Essbase Technical Reference*

Because unnecessary blocks can be created when Create Blocks on Equations is enabled at the application or database level, calculation performance can be affected. To control block creation within a calculation script, use the SET CREATEBLOCKONEQ ON | OFF calculation command. See "Nonconstant Values Assigned to Members in a Sparse Dimension" on page 877.

Understanding Formula Calculation

For formulas applied to members in a database outline, Essbase calculates formulas when you perform the following actions:

- Run a default (CALC ALL) calculation of a database.
- Run a calculation script that calculates the member containing the formula; for example, a CALC DIM of the dimension containing the member, or the member itself. See Chapter 28, "Developing Calculation Scripts."

For a formula in a calculation script, Essbase calculates the formula when it occurs in the calculation script.

If a formula is associated with a dynamically calculated member, Essbase calculates the formula when the user requests the data values. In a calculation script, you cannot calculate a dynamically calculated member or make a dynamically calculated member the target of a formula calculation. See Chapter 26, "Dynamically Calculating Data Values."

Using dynamically calculated members in a formula on a database outline or in a calculation script can significantly affect calculation performance. Performance is affected because Essbase has to interrupt the regular calculation to perform the dynamic calculation.

You cannot use substitution variables in formulas that you apply to the database outline. See "Using Substitution Variables in Formulas" on page 367.

Understanding Formula Syntax

When you create member formulas, follow these rules:

- End each statement in the formula with a semicolon (;). For example,

  ```
  Margin % Sales;
  ```

- Use only saved outline member names. If a substitution variable is used for a member name, the substitution variable value must be a saved outline member name.
- Enclose a member name in double quotation marks (`" "`) if the member name meets any of the following conditions:
  - Contains spaces. For example:

```
"Opening Inventory" = "Ending Inventory" - Sales + Additions;
```

- ❍ Is the same as an operator, function name, or keyword. See "Using Dimension and Member Names in Calculation Scripts, Report Scripts, Formulas, Filters, Substitution Variable Values and Environment Variable Values" on page 1063.
- ❍ Includes any nonalphanumeric character. For example, hyphens (-), asterisks (*), and slashes (/).
- ❍ Is all numeric or starts with one or more numerals. For example, `"100"` or `"10Prod"`

For a full list of member names that must be enclosed in quotation marks, see "Using Dimension and Member Names in Calculation Scripts, Report Scripts, Formulas, Filters, Substitution Variable Values and Environment Variable Values" on page 1063.

- End each IF statement in a formula with an ENDIF statement.

  For example, the following formula contains a simple IF...ENDIF statement. You can apply this formula to the Commission member in a database outline:

  ```
  IF(Sales < 100)
     Commission = 0;
  ENDIF;
  ```

  If you are using an IF statement nested within another IF statement, end each IF with an ENDIF. For example:

  ```
  "Opening Inventory"
  (IF (@ISMBR(Budget))
     IF (@ISMBR(Jan))
     "Opening Inventory" = Jan;
     ELSE
     "Opening Inventory" = @PRIOR("Ending Inventory");
     ENDIF;
  ENDIF;)
  ```

- You need not end ELSE or ELSEIF statements with ENDIFs. For example:

  ```
  IF (@ISMBR(@DESCENDANTS(West)) OR @ISMBR(@DESCENDANTS(East)
     Marketing = Marketing * 1.5;
  ELSEIF(@ISMBR(@DESCENDANTS(South)))
     Marketing = Marketing * .9;
  ELSE Marketing = Marketing * 1.1;
  ENDIF;
  ```

  Note:

  If you use ELSE IF (with a space) rather than ELSEIF (one word) in a formula, you must supply an ENDIF for the IF statement.

- Ending ENDIF statements with a semicolon (;) is not required, but it is a good practice.

When writing formulas, you can check the syntax using the Formula Editor syntax checker. See "Checking Formula Syntax" on page 376.

For information on syntax for Essbase functions and commands, see the *Oracle Essbase Technical Reference*.

Reviewing the Process for Creating Formulas

You use Formula Editor, a tab in the Member Properties dialog box in Outline Editor, to create formulas. You can type the formulas directly into the formula text area, or you can use the Formula Editor UI features to create the formula.

Formulas are plain text. If required, you can create a formula in the text editor of your choice and paste it into Formula Editor.

To create a formula:

1. In Outline Editor, select the member to which to apply the formula.
2. Open Formula Editor.

   See "Creating and Editing Formulas in Outlines" in the *Oracle Essbase Administration Services Online Help*.
3. Enter the formula text.

   See "Composing Formulas" on page 359 and "Creating and Editing Formulas in Outlines" in the *Oracle Essbase Administration Services Online Help*.
4. Check the formula syntax.

   See "Checking Formula Syntax" on page 376.
5. Save the formula.

   See "Creating and Editing Formulas in Outlines" in the *Oracle Essbase Administration Services Online Help*.
6. Save the outline.

   See "Saving Outlines" in the *Oracle Essbase Administration Services Online Help*.

Displaying Formulas

➤ To display a formula, use a tool:

Tool	Topic	Location
Administration Services	Creating and Editing Formulas in Outlines	*Oracle Essbase Administration Services Online Help*
ESSCMD	GETMBRCALC	*Oracle Essbase Technical Reference*
MaxL	**query database**	*Oracle Essbase Technical Reference*

Composing Formulas

The topics in this section discuss the main formula types. Also see Chapter 23, "Reviewing Examples of Formulas."

Before writing formulas, review the guidelines in "Understanding Formula Syntax" on page 357.

Basic Equations

You can apply a mathematical operation to a formula to create a basic equation. The equation can be in the database outline or in a calculation script.

In a calculation script, you define basic equations as:

```
Member = mathematical operation;
```

where `Member` is a member name from the database outline and `mathematical operation` is any valid mathematical operation. For example:

```
Margin = Sales - COGS;
```

In this example, Essbase cycles through the database subtracting the values in COGS from the values in Sales and placing the results in Margin.

As another example, you can apply the following formula to a Markup member:

```
(Retail - Cost) % Retail;
```

In a calculation script, this formula is:

```
Markup = (Retail - Cost) % Retail;
```

In this example, Essbase cycles through the database subtracting the values in Cost from the values in Retail, calculating the resulting values as a percentage of the values in Retail, and placing the result in Markup.

Conditional Tests

You can define formulas that use a conditional test or a series of conditional tests to control the flow of calculation.

The IF and ENDIF commands define a *conditional block*. The formulas between the IF and the ENDIF commands are executed only if the test returns TRUE (1). If the test returns FALSE (0), you can use the ELSE and ELSEIF commands to specify alternative actions. The formulas following each ELSE command are executed only if the previous test returns FALSE (0). Conditions following each ELSEIF command are tested only if the previous IF command returns FALSE (0). See "Understanding Formula Syntax" on page 357.

When you use a conditional formula in a calculation script, enclose it in parentheses and associate it with a member in the database outline, as shown in the examples in this section.

In conjunction with an IF command, you can use functions that return TRUE or FALSE (1 or 0, respectively) based on the result of a conditional test. These functions are known as *Boolean functions*.

Use Boolean functions to determine which formula to use. The decision is based on the characteristics of the current member combination. For example, to restrict a certain calculation to the members in the Product dimension that contain input data, preface the calculation with an IF test based on @ISLEV(Product,0).

If one of the function parameters is a cross-dimensional member, such as @ISMBR(Sales -> Budget), all of the parts of the cross-dimensional member must match the properties of the current cell to return a value of TRUE (1).

You can use the following Boolean functions to specify conditions:

Condition	Function
Current member has a specified accounts tag (for example, an Expense tag)	@ISACCTYPE
Current member is an ancestor of the specified member	@ISANCEST
Current member is an ancestor of the specified member, or the specified member itself	@ISIANCEST
Current member is a child of the specified member	@ISCHILD
Current member is a child of the specified member, or the specified member itself	@ISICHILD
Current member is a descendant of the specified member	@ISDESC
Current member is a descendant of the specified member, or the specified member itself	@ISIDESC
Current member of the specified dimension is in the generation specified	@ISGEN
Current member of the specified dimension is in the level specified	@ISLEV
Current member matches any of the specified members	@ISMBR
Current member is the parent of the specified member	@ISPARENT
Current member is the parent of the specified member, or the specified member itself	@ISIPARENT
Current member (of the same dimension as the specified member) is in the same generation as the specified member	@ISSAMEGEN
Current member (of the same dimension as the specified member) is in the same level as the specified member	@ISSAMELEV
Current member is a sibling of the specified member	@ISSIBLING
Current member is a sibling of the specified member, or the specified member itself	@ISISIBLING
A specified UDA exists for the current member of the specified dimension	@ISUDA

When you place formulas on the database outline, you can use only the IF, ELSE, ELSEIF, and ENDIF commands and Boolean functions to control the flow of the calculations. You can use additional control commands in a calculation script.

For information about how to develop calculation scripts and how to use them to control how Essbase calculates a database, see Chapter 28, “Developing Calculation Scripts.” For information on individual Essbase functions and calculation commands, see the *Oracle Essbase Technical Reference*.

Examples of Conditional Tests

You can apply the following formula to a Commission member in the database outline.

In the following example, the formula calculates commission at 1% of sales if the sales are greater than 500000:

```
IF(Sales > 500000)
   Commission = Sales * .01;
ENDIF;
```

If you place the formula in a calculation script, you must associate the formula with the Commission member as shown:

```
Commission (IF(Sales > 500000)
   Commission = Sales * .01;
ENDIF;)
```

Essbase cycles through the database, performing these calculations:

1. The IF statement checks to see if the value of Sales for the current member combination is greater than 500000.
2. If Sales is greater than 500000, Essbase multiplies the value in Sales by 0.01 and places the result in Commission.

In the next example, the formula tests the ancestry of the current member and then applies the appropriate Payroll calculation formula:

```
IF(@ISIDESC(East) OR @ISIDESC(West))
   Payroll = Sales * .15;
ELSEIF(@ISIDESC(Central))
   Payroll = Sales * .11;
ELSE
   Payroll = Sales * .10;
ENDIF;
```

If you place the formula in a calculation script, you must associate the formula with the Payroll member as shown:

```
Payroll(IF(@ISIDESC(East) OR @ISIDESC(West))
   Payroll = Sales * .15;
ELSEIF(@ISIDESC(Central))
   Payroll = Sales * .11;
ELSE
   Payroll = Sales * .10;
ENDIF;)
```

Essbase cycles through the database, performing the following calculations:

1. The IF statement uses the @ISIDESC function to check whether the current member on the Market dimension is a descendant of either East or West.
2. If the current member on the Market dimension is a descendant of East or West, Essbase multiplies the value in Sales by 0.15 and moves on to the next member combination.
3. If the current member is not a descendant of East or West, the ELSEIF statement uses the @ISIDESC function to check whether the current member is a descendant of Central.

4. If the current member on the Market dimension is a descendant of Central, Essbase multiplies the value in Sales by 0.11 and moves to the next member combination.
5. If the current member is not a descendant of East, West, or Central, Essbase multiplies the value in Sales by 0.10 and moves to the next member combination.

See "About Multidimensional Calculation Concepts" on page 347. For information on the @ISIDESC function, see the *Oracle Essbase Technical Reference*.

Value-Related Formulas

Use this section to find information about formulas related to values.

Using Interdependent Values

Essbase optimizes calculation performance by calculating formulas for a range of members in the same dimension at the same time. Some formulas, however, require values from members of the same dimension, and Essbase may not yet have calculated the required values.

A good example is that of cash flow, in which the opening inventory is dependent on the ending inventory from the previous month.

In Sample.Basic, the Opening Inventory and Ending Inventory values must be calculated on a month-by-month basis.

	Jan	Feb	Mar
Opening Inventory	100	120	110
Sales	50	70	100
Addition	70	60	150
Ending Inventory	120	110	160

Assuming that the Opening Inventory value for January is loaded into the database, the required calculation:

```
1. January Ending   = January Opening - Sales + Additions
2. February Opening = January Ending
3. February Ending  = February Opening - Sales + Additions
4. March Opening    = February Ending
5. March Ending     = March Opening - Sales + Additions
```

You can calculate the required results by applying interdependent, multiple equations to one member in the database outline.

The following formula, applied to the Opening Inventory member in the database outline, calculates the correct values:

```
IF(NOT @ISMBR (Jan))
    "Opening Inventory" = @PRIOR("Ending Inventory");
ENDIF;
"Ending Inventory" = "Opening Inventory" - Sales + Additions;
```

If you place the formula in a calculation script, you must associate the formula with the Opening Inventory member as shown:

```
"Opening Inventory"
(IF(NOT @ISMBR (Jan))
   "Opening Inventory" = @PRIOR("Ending Inventory");
ENDIF;)
"Ending Inventory" = "Opening Inventory" - Sales + Additions;
```

Essbase cycles through the months, performing the following calculations:

1. The IF statement and @ISMBR function check that the current member on the Year dimension is not Jan. This step is necessary because the Opening Inventory value for Jan is an input value.
2. If the current month is not Jan, the @PRIOR function obtains the value for the Ending Inventory for the previous month. This value is then allocated to the Opening Inventory of the current month.
3. The Ending Inventory is calculated for the current month.

Note:

To calculate the correct results, you must place the above formula on one member, Opening Inventory. If you place the formulas for Opening Inventory and Ending Inventory on their separate members, Essbase calculates Opening Inventory for all months and then Ending Inventory for all months. This organization means that the value of the Ending Inventory of the previous month is not available when Opening Inventory is calculated.

Calculating Variances or Percentage Variances Between Actual and Budget Values

You can use the @VAR and @VARPER functions to calculate a variance or percentage variance between budget and actual values.

You may want the variance to be positive or negative, depending on whether you are calculating variance for members on the accounts dimension that are expense or nonexpense items:

- Expense items. You want Essbase to show a positive variance if the actual values are less than the budget values (for example, if actual costs are less than budgeted costs).
- Nonexpense items. You want Essbase to show a negative variance if the actual values are less than the budget values (for example, if actual sales are less than budgeted sales).

By default, Essbase assumes that members are nonexpense items and calculates the variance accordingly.

➤ To tell Essbase that a member is an expense item:

1 **In Outline Editor, select the member.**

The member must be on the dimension tagged as accounts.

2 **Open Formula Editor.**

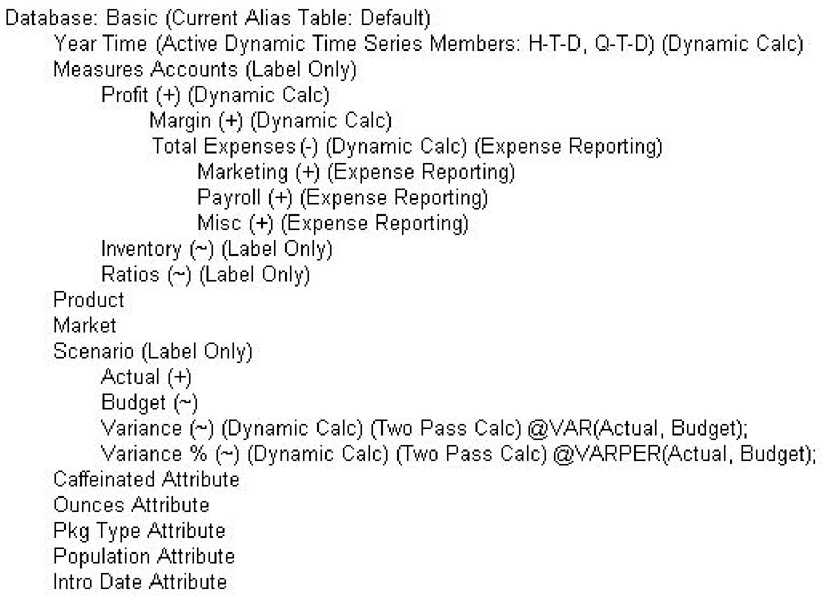
See “Creating and Editing Formulas in Outlines” in the *Oracle Essbase Administration Services Online Help*.

3 **Tag the member as an expense item.**

See “Setting Variance Reporting Properties” on page 144.

When you use the @VAR or @VARPER functions, Essbase shows a positive variance if the actual values are less than the budget values. For example, in Sample.Basic, the children of Total Expenses are expense items. The Variance and Variance % members of the Scenario dimension calculate the variance between the Actual and Budget values.

Figure 115 Sample.Basic Showing Expense Items



```
Database: Basic (Current Alias Table: Default)
    Year Time (Active Dynamic Time Series Members: H-T-D, Q-T-D) (Dynamic Calc)
    Measures Accounts (Label Only)
        Profit (+) (Dynamic Calc)
            Margin (+) (Dynamic Calc)
            Total Expenses (-) (Dynamic Calc) (Expense Reporting)
                Marketing (+) (Expense Reporting)
                Payroll (+) (Expense Reporting)
                Misc (+) (Expense Reporting)
        Inventory (~) (Label Only)
        Ratios (~) (Label Only)
    Product
    Market
    Scenario (Label Only)
        Actual (+)
        Budget (~)
        Variance (~) (Dynamic Calc) (Two Pass Calc) @VAR(Actual, Budget);
        Variance % (~) (Dynamic Calc) (Two Pass Calc) @VARPER(Actual, Budget);
    Caffeinated Attribute
    Ounces Attribute
    Pkg Type Attribute
    Population Attribute
    Intro Date Attribute
```

Allocating Values

Allocation functions allow you to allocate values that are input at the parent level across child members in the same dimension or in different dimensions. The allocation is based on a variety of specified criteria.

Allocated Values	Function
Values from a member, cross-dimensional member, or value across a member list within the same dimension.	@ALLOCATE
Values from a member, cross-dimensional member, or value across multiple dimensions.	@MDALLOCATE

For examples of calculation scripts using @ALLOCATE, see “Allocating Costs Across Products” on page 475; using @MDALLOCATE, see “Allocating Values Across Multiple Dimensions” on page 478.

Forecasting Values

Forecasting functions allow you to manipulate data for the purposes of interpolating data or calculating future values.

Data Manipulation	Function
Apply a moving average to a data set and replace each term in the list with a trailing average. This function modifies the data set for smoothing purposes.	@MOVAVG
Apply a moving maximum to a data set and replace each term in the list with a trailing maximum. This function modifies the data set for smoothing purposes.	@MOVMAX
Apply a moving median to a data set and replace each term in the list with a trailing median. This function modifies the data set for smoothing purposes.	@MOVMED
Apply a moving minimum to a data set and replace each term in the list with a trailing minimum. This function modifies the data set for smoothing purposes.	@MOVMIN
Apply a moving sum to a data set and replace each term with a trailing sum. This function modifies the data set for smoothing purposes.	@MOVSUM
Apply a moving sum to a data set and replace each term with a trailing sum. Specify how to assign values to members before you reach the number to sum. This function modifies the data set for smoothing purposes.	@MOVSUMX
Apply a smoothing spline to a set of data points. A spline is a mathematical curve that is used to smooth or interpolate data.	@SPLINE
Calculate future values and base the calculation on curve-fitting to historical values.	@TREND

For information about specific Essbase functions, see the *Oracle Essbase Technical Reference*.

Using Member Relationships to Look Up Values

Relationship functions allow you to use the member combination that Essbase is currently calculating to look up specific values.

Look-up Value	Function
Ancestor values of the specified member combination	@ANCESTVAL
Numeric value of the attribute from the specified numeric or date attribute dimension associated with the current member	@ATTRIBUTEVAL
Text value of the attribute from the specified text attribute dimension associated with the current member	@ATTRIBUTESVAL
Value (TRUE or FALSE) of the attribute from the specified Boolean attribute dimension associated with the current member	@ATTRIBUTEBVAL

Look-up Value	Function
Generation number of the current member combination for the specified dimension	@CURGEN
Level number of the current member combination for the specified dimension	@CURLEV
Generation number of the specified member	@GEN
Level number of the specified member	@LEV
Ancestor values of the specified member combination across multiple dimensions	@MDANCESTVAL
Shared ancestor values of the specified member combination	@SANCESTVAL
Parent values of the specified member combination	@PARENTVAL
Parent values of the specified member combination across multiple dimensions	@MDPARENTVAL
Shared parent values of the specified member combination	@SPARENTVAL
Data value from another database to be used for calculation of a value from the current database	@XREF

For information about specific Essbase functions, see the *Oracle Essbase Technical Reference*.

Using Substitution Variables in Formulas

Substitution variables act as placeholders for information that changes regularly; for example, time-period information. You can use substitution variables in formulas that you apply to the database outline.

When the outline is calculated, Essbase replaces the substitution variable with the value that you have assigned to it. You can create and assign values to substitution variables using Administration Services, MaxL, or ESSCMD.

You can set substitution variables at the server, application, and database levels. Essbase must be able to access the substitution variable from the application and database on which you are running the calculation scripts. See "Using Substitution Variables" on page 120.

To use a substitution variable in a formula, enter an ampersand (`&`), followed by the substitution variable name.

Essbase treats any text string preceded by `&` as a substitution variable.

For example, assume that the substitution variable UpToCurr is defined as Jan:Jun. You can use the following @ISMBR function as part of a conditional test:

```
@ISMBR(&UpToCurr)
```

At the time Essbase calculates the outline, it replaces the substitution variable, as shown:

```
@ISMBR(Jan:Jun)
```

Note:

Substitution variables used in formulas for new outline members do not pass verification unless the outline is saved.

Using Environment Variables in Formulas

In outline member formulas, you can use system environment variables as placeholders for user-specific system settings. Because environment variables are defined at the operating system level, they are available to all formulas on Essbase Server.

Using environment variables in formulas is the same as using them in calculation scripts. See "Using Environment Variables in Calculation Scripts" on page 456.

Note:

Environment variables cannot be used in MDX queries or in member formulas that are within aggregate storage outlines.

Member-Related Formulas

This section provides information about creating formulas that refer to members.

Specifying Member Lists and Ranges

In some functions, you may need to specify more than one member, or you may need to specify a range of members. For example, the @ISMBR function tests to see if a member that is currently being calculated matches any of a list or range of specified members.

You can specify members using the following syntax:

Member List or Range	Syntax
One member	The member name. For example: Mar2001
A list of members	A comma-delimited (,) list of member names. For example: Mar2001, Apr2001, May2001
A range of all members at the same level, between and including the two defining members	The two defining member names separated by a colon (:). For example: Jan2000:Dec2000
A range of all members in the same generation, between and including the two defining members	The two defining member names separated by two colons (::). For example: Q1_2000::Q4_2000
A function-generated list of members or a range of members	For a list of member list contents and corresponding functions, see "Generating Member Lists" on page 369.
A combination of ranges and list	Separate each range, list, and function with a comma (,). For example:

Member List or Range	Syntax
	Q1_97::Q4_98, FY99, FY2000 or @SIBLINGS(Dept01), Dept65:Dept73, Total_Dept

If you do not specify a list of members or a range of members in a function that requires either, Essbase uses the level 0 members of the dimension tagged as time. If no dimension is tagged as time, Essbase displays an error message.

Generating Member Lists

Member set functions allow you to generate member lists that are based on a specified member or member list.

Contents of Member List	Function
All ancestors of the specified member, including ancestors of the specified member as a shared member. This function does not include the specified member.	@ALLANCESTORS
All ancestors of the specified member, including ancestors of the specified member as a shared member. This function includes the specified member.	@IALLANCESTORS
The ancestor of the specified member at the specified generation or level	@ANCEST
All ancestors of the specified member (optionally, up to the specified generation or level), but not the specified member	@ANCESTORS
All ancestors of the specified member (optionally, up to the specified generation or level), including the specified member	@IANCESTORS
All ancestors of the specified list of members (optionally, up to the specified generation or level), but not including the specified members	@LANCESTORS
All ancestors of the specified list of members (optionally, up to the specified generation or level), including the specified members	@ILANCESTORS
All children of the specified member, but not including the specified member	@CHILDREN
All children of the specified member, including the specified member	@ICHILDREN
The current member being calculated for the specified dimension	@CURRMBR
All descendants of the specified member (optionally, up to the specified generation or level), but not the specified member nor descendants of shared members	@DESCENDANTS
All descendants of the specified member (optionally, up to the specified generation or level), including the specified member, but not descendants of shared members	@IDESCENDANTS
All descendants of the specified list of members (optionally, down to the specified generation or level), but not including the specified members	@LDESCENDANTS
All descendants of the specified list of members (optionally, down to the specified generation or level), including the specified members	@ILDESCENDANTS

Contents of Member List	Function
All descendants of the specified member (optionally, up to the specified generation or level), including descendants of shared members, but not the specified member	@RDESCENDANTS
All descendants of the specified member (optionally, up to the specified generation or level), including the specified member and descendants of shared members	@IRDESCENDANTS
All members of the specified generation in the specified dimension	@GENMBRS
All members of the specified level in the specified dimension	@LEVMBRS
All siblings of the specified member, but not the specified member	@SIBLINGS
All siblings of the specified member, including the specified member	@ISIBLINGS
All siblings that precede the specified member in the database outline, but not the specified member	@LSIBLINGS
All siblings that follow the specified member in the database outline, but not the specified member	@RSIBLINGS
All siblings that precede the specified member in the database outline, including the specified member	@ILSIBLINGS
All siblings that follow the specified member in the database outline, including the specified member	@IRSIBLINGS
The sibling at the specified distance from the member.	@SHIFTSIBLING
The next, or right-most, sibling of the member.	@NEXTSIBLING
The previous, or left-most, sibling of the member.	@PREVSIBLING
Separate lists of members to be processed by functions that require multiple list arguments	@LIST
The member with the name that is provided as a character string	@MEMBER
A merged list of two member lists to be processed by another function	@MERGE
A member list that crosses the specified member from one dimension with the specified member range from another dimension	@RANGE
A member list that identifies all shared members among the specified members	@SHARE
A member list that identifies the range of members between (and inclusive of) two specified single or cross-dimensional members at the same level	@XRANGE
A list of members from which some members have been removed	@REMOVE
All members that match the specified wildcard selection	@MATCH
The parent of the current member being calculated in the specified dimension	@PARENT
All members of the specified generation or level that are above or below the specified member	@RELATIVE
All members that have a common UDA defined on Essbase Server	@UDA

Contents of Member List	Function
All base-dimension members that are associated with the specified attribute-dimension member	@ATTRIBUTE
All base members that are associated with attributes that satisfy the specified conditions	@WITHATTR

For information about specific Essbase functions, see the *Oracle Essbase Technical Reference*.

Manipulating Member Names

You can work with member names as character strings by using the following functions:

Character String Manipulation	Function
Create a character string that is the result of appending a member name or specified character string to another member name or character string	@CONCATENATE
Return a member name as a string	@NAME
Return a substring of characters from another character string or from a member name	@SUBSTRING

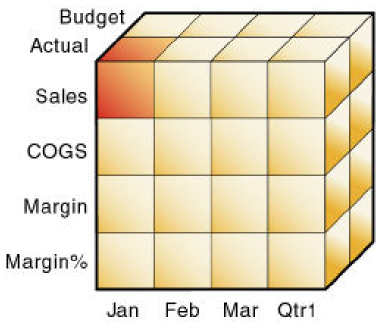
Working with Member Combinations Across Dimensions

Use the cross-dimensional operator to point to data values of specific member combinations. Create the cross-dimensional operator using a hyphen (-) and a greater-than symbol (>). Do not include a space between the cross-dimensional operator and members.

In the following simplified illustration, the shaded data value is

Sales -> Jan -> Actual

Figure 116 Defining a Single Data Value by Using the Cross-Dimensional Operator



The following example, which allocates miscellaneous expenses to each product in each market, illustrates how to use the cross-dimensional operator. The value of Misc_Expenses for all products in all markets is known. The formula allocates a percentage of the total Misc_Expenses value to each Product -> Market combination. The allocation is based on the value of Sales for each product in each market.

```
Misc_Expenses = Misc_Expenses -> Market -> Product *
```

```
   (Sales / ( Sales -> Market -> Product));
```

Essbase cycles through the database, performing these calculations:

1. Essbase divides the Sales value for the current member combination by the total Sales value for all markets and all products (Sales -> Market -> Product).
2. It multiplies the value calculated in step 1 by the Misc_Expenses value for all markets and all products (Misc_Expenses -> Market -> Product).
3. It allocates the result to Misc_Expenses for the current member combination.

Using the cross-dimensional operator can have significant performance implications. For optimization guidelines, see "Using Cross-Dimensional Operators" on page 878.

Formulas That Use Various Types of Functions

The topics in this section discuss formulas that use other types of functions. For more information about specific Essbase functions, see the *Oracle Essbase Technical Reference*.

Mathematical Operations

Mathematical functions allow you to perform many mathematical operations in formulas.

Operation	**Function**
Return the absolute value of an expression	@ABS
Return the average value of the values in the specified member list	@AVG
Return the value of e (the base of natural logarithms) raised to power of the specified expression	@EXP
Return the factorial of an expression	@FACTORIAL
Return the next-lowest integer value of a member or expression	@INT
Return the natural logarithm of a specified expression	@LN
Return the logarithm to a specified base of a specified expression	@LOG
Return the base-10 logarithm of a specified expression	@LOG10
Return the maximum value among the expressions in the specified member list	@MAX
Return the maximum value among the expressions in the specified member list, with the ability to skip zero and #MISSING values	@MAXS
Return the minimum value among the expressions in the specified member list	@MIN
Return the minimum value among the expressions in the specified member list, with the ability to skip zero and #MISSING values	@MINS
Return the modulus produced by the division of two specified members	@MOD

Operation	Function
Return the value of the specified member raised to the specified power	@POWER
Return the remainder value of an expression	@REMAINDER
Return the member or expression rounded to the specified number of decimal places	@ROUND
Return the summation of values of all specified members	@SUM
Return the truncated value of an expression	@TRUNCATE
Return the variance (difference) between two specified members. See "Calculating Variances or Percentage Variances Between Actual and Budget Values" on page 364.	@VAR
Return the percentage variance (difference) between two specified members. See "Calculating Variances or Percentage Variances Between Actual and Budget Values" on page 364.	@VARPER

Statistical Functions

Statistical functions allow you to calculate advanced statistics in Essbase.

Calculated Value	Function
The correlation coefficient between two parallel data sets	@CORRELATION
The number of values in the specified data set	@COUNT
The median, or middle number, in the specified data set	@MEDIAN
The mode, or the most frequently occurring value, in the specified data set	@MODE
The rank of the specified member or value in the specified data set	@RANK
The standard deviation, based upon a sample, of the specified members	@STDEV
The standard deviation, based upon the entire population, of the specified members	@STDEVP
The standard deviation, crossed with a range of members, of the specified members	@STDEVRANGE
The variance, based upon a sample, of the specified data set	@VARIANCE
The variance, based upon the entire population, of the specified data set	@VARIANCEP

Range Functions

Range functions allow you to execute a function for a range of members.

Calculation	Function
The average value of a member across a range of members	@AVGRANGE
A range of members that is based on the relative position of the member combination Essbase is currently calculating	@CURRMBRRANGE
The maximum value of a member across a range of members	@MAXRANGE
The maximum value of a member across a range of members, with the ability to skip zero and #MISSING values	@MAXSRANGE
The next or *n*th member in a range of members, retaining all other members identical to the current member across multiple dimensions	@MDSHIFT
The minimum value of a member across a range of members	@MINRANGE
The minimum value of a member across a range of members, with the ability to skip zero and #MISSING values	@MINSRANGE
The next or *n* th member in a range of members	@NEXT
The next or *n*th member in a range of members, with the option to skip #MISSING, zero, or both values	@NEXTS
The previous or *n*th previous member in a range of members	@PRIOR
The previous or *n*th previous member in a range of members, with the option to skip #MISSING, zero, or both values	@PRIORS
The next or *n*th member in a range of members, retaining all other members identical to the current member and in the specified dimension	@SHIFT In some cases, @SHIFTPLUS or @SHIFTMINUS.
The summation of values of all specified members across a range of members	@SUMRANGE

Financial Functions

Financial functions allow you to include financial calculations in formulas.

Calculation	Function
An accumulation of values up to the specified member	@ACCUM
The proceeds of a compound interest calculation	@COMPOUND
A series of values that represent the compound growth of the specified member across a range of members	@COMPOUNDGROWTH
Depreciation for a specific period, calculated using the declining balance method	@DECLINE
A value discounted by the specified rate, from the first period of the range to the period in which the amount to discount is found	@DISCOUNT

Calculation	Function
A series of values that represents the linear growth of the specified value	@GROWTH
The simple interest for a specified member at a specified rate	@INTEREST
The internal rate of return on a cash flow	@IRR
The Net Present Value of an investment (based on a series of payments and incomes)	@NPV
The period-to-date values of members in the dimension tagged as time	@PTD
The amount per period that an asset in the current period may be depreciated (calculated across a range of periods). The depreciation method used is straight-line depreciation.	@SLN
The amount per period that an asset in the current period may be depreciated (calculated across a range of periods). The depreciation method used is sum of the year's digits.	@SYD

Note:

One member formula cannot contain multiple financial functions (for example, @NPV and @SLN, or multiple instances of @NPV). A member formula that requires multiple financial functions must be broken into separate formulas so that each formula contains only one financial function (for example, *MemberName*(@NPV(...));*Membername*(@NPV(...))).

Date and Time Function

The date function allows you to use dates with other functions.

Date Conversion	Function
Convert date strings to numbers that can be used in calculation formulas	@TODATE

Calculation Mode Function

The calculation mode function allows you to specify which calculation mode that Essbase uses to calculate a formula.

Calculation Mode	Function
Specify the calculation mode (cell, block, bottom-up, or top-down) that Essbase uses to calculate a formula	@CALCMODE

Note:

You can also use the configuration setting CALCMODE to set calculation modes to BLOCK or BOTTOMUP at the database, application, or server level. See the *Oracle Essbase Technical Reference*.

Custom-Defined Functions

You can create custom-defined functions, to be used in formulas and calculation scripts, to perform calculations not otherwise supported by the Essbase calculation scripting language. Custom-developed functions must be written in the Java programming language and registered on the Essbase Server. The Essbase calculator framework calls them as external functions.

Custom-defined functions are displayed in the functions tree in Calculation Script Editor, where you can select them to insert into a formula.

See Chapter 31, "Developing Custom-Defined Calculation Functions."

Checking Formula Syntax

Essbase includes Essbase Server-based formula syntax checking that tells you about syntax errors in formulas. For example, Essbase tells you if you have mistyped a function name. Unknown names can be validated against a list of custom-defined macro and function names. If you are not connected to a server or the application associated with the outline, Essbase may connect you to validate unknown names.

A syntax checker cannot tell you about semantic errors in a formula. Semantic errors occur when a formula does not work as you expect. To find semantic errors, run the calculation and check the results to ensure that they are as you expect.

Essbase displays the syntax checker results at the bottom of the Formula Editor. If Essbase finds no syntax errors, it displays the "No errors" message.

If Essbase finds one or more syntax errors, it displays the number of the line that includes the error and a brief description of the error. For example, if you do not include a semicolon end-of-line character at the end of a formula, Essbase displays a message similar to the following message:

```
Error: line 1: invalid statement; expected semicolon
```

If a formula passes validation in Formula Editor or Outline Editor, but Essbase Server detects semantic errors when the outline is saved, check the following:

- The incorrect formula is saved as part of the outline, even though it contains errors.
- Essbase Server writes a message in the application log that indicates what the error is and displays the incorrect formula.
- Essbase Server writes an error message to the comment field of the member associated with the incorrect formula. The message indicates that the incorrect formula was not loaded. You can view this comment in Outline Editor by closing and reopening the outline.
- If you do not correct the member formula, and a calculation that includes that member is run, the formula is ignored during the calculation.

After you have corrected the formula and saved the outline, the message in the member comment is deleted. You can view the updated comment when you reopen the outline.

➤ To check formula syntax, see "Creating and Editing Formulas in Outlines" in *Oracle Essbase Administration Services Online Help*.

Estimating Disk Size for a Calculation

You can estimate the disk size required for a single CALC ALL for a full data load or a partial data load. See "Estimating Calculation Affects on Database Size" on page 865.

➤ To estimate disk size for a calculation, see ESTIMATEFULLDBSIZE in the *Oracle Essbase Technical Reference*.

Using Formulas in Partitions

An Essbase partition can span multiple Essbase Servers, processors, or computers. See Chapter 14, "Designing Partitioned Applications" and Chapter 15, "Creating and Maintaining Partitions".

You can use formulas in partitioning, just as you use formulas on your local database. If, however, a formula you use in one database references a value from another database, Essbase has to retrieve the data from the other database when calculating the formula; therefore, ensure that the referenced values are up-to-date and carefully consider the performance impact on the overall database calculation. See "Writing Calculation Scripts for Partitions" on page 466.

With transparent partitions, carefully consider how you use formulas on the data target. See "Transparent Partitions and Member Formulas" on page 229 and "Performance Considerations for Transparent Partitions" on page 227.

23 Reviewing Examples of Formulas

In This Chapter

The information in this chapter applies only to block storage databases and is not relevant to aggregate storage databases.

Also see:

- Chapter 57, "Comparison of Aggregate and Block Storage"
- "Developing Formulas on Aggregate Storage Outlines" on page 930
- Chapter 29, "Reviewing Examples of Calculation Scripts"

Calculating Period-to-Date Values

If the outline includes a dimension tagged as accounts, you can use the @PTD function to calculate period-to-date values.

The example in this topic uses the Inventory branch of the Measures dimension from the Sample.Basic database, as shown:

```
Inventory (~) (Label Only)
   Opening Inventory (+) (TB First) (Expense Reporting) IF(NOT @ISMBR(Jan))
   Additions (~) (Expense Reporting)
   Ending Inventory (~) (TB Last) (Expense Reporting)
```

To calculate period-to-date values for the year and for the current quarter, add two members to the Year dimension: QTD for quarter-to-date and YTD for year-to-date. For example:

```
QTD (~) @PTD(Apr:May)
YTD (~) @PTD(Jan:May);
```

Assuming that the current month is May, you would add this formula to the QTD member:

```
@PTD(Apr:May);
```

And you would add this formula on the YTD member:

```
@PTD(Jan:May);
```

Essbase sums the values for the range of months, as appropriate. Opening Inventory, however, has a time balance tag, First, and Ending Inventory has a time balance tag, Last. Essbase takes these values and treats them accordingly. See "Calculating First, Last, and Average Values" on page 435.

The following table provides an example of the calculation results for the members in the Inventory branch and for the Sales member:

Measures -> Time	Jan	Feb	Mar	Apr	May	QTD	YTD
Opening Inventory	100	110	120	110	140	**110**	**100**
Additions	110	120	100	160	180	**340**	**670**
Sales	100	110	110	130	190	**320**	**640**
Ending Inventory	110	120	110	140	130	**130**	**130**

The values for Sales and Additions have been summed.

Opening Inventory has a First tag. For QTD, Essbase takes the first value in the current quarter, which is Apr. For YTD, Essbase takes the first value in the year, which is Jan.

Ending Inventory has a Last tag. For QTD, Essbase takes the last value in the current quarter, which is May. For YTD, Essbase takes the last value in the year, which is also May.

Note:

You can also use Dynamic Time Series members to calculate period-to-date values. See Chapter 27, "Calculating Time Series Data."

Calculating Rolling Values

You can use the @AVGRANGE function to calculate rolling averages and the @ACCUM function to calculate rolling year-to-date values.

For example, assume that a database contains monthly Sales data values and that the database outline includes the members AVG_Sales and YTD_Sales.

You would add this formula to the AVG_Sales member:

```
@AVGRANGE(SKIPNONE, Sales, @CURRMBRRANGE(Year, LEV, 0, , 0));
```

And you would add this formula on the YTD_Sales member:

```
@ACCUM(Sales);
```

Essbase calculates the average Sales values across the months in the dimension tagged as time. The SKIPNONE parameter means that all values are included, even #MISSING values. Essbase places the results in AVG_Sales. See "Consolidating #MISSING Values" on page 894.

The following table shows the results when Essbase calculates the cumulative Sales values and places the results in YTD_Sales:

Measures -> Time	Jan	Feb	Mar	Qtr1
Sales	100	200	300	**600**
AVG_Sales	100	150	200	#MISSING
YTD_Sales	100	300	600	#MISSING

The values for AVG_Sales are averages of the months-to-date. For example, AVG_Sales -> Mar is an average of Sales for Jan, Feb, and Mar.

The values for YTD_Sales are the cumulative values up to the current month. So YTD_Sales -> Feb is the sum of Sales -> Jan and Sales -> Feb.

Calculating Monthly Asset Movements

You can use the @PRIOR function to calculate values based on a previous month's value.

For example, assume that a database contains assets data values that are stored on a month-by-month basis. You can calculate the difference between the assets values of successive months (the asset movement) by subtracting the previous month's value from the present month's value.

Assume these three members manage the asset values for the database:

- Assets for the monthly asset values
- Asset_MVNT for the asset movement values
- Opening_Balance for the asset value at the beginning of the year

For Jan, the Asset_MVNT value is calculated by subtracting the Opening_Balance value from the Jan value.

You would add this formula on the Asset_MVNT member:

```
IF(@ISMBR(Jan)) Asset_MVNT = Assets - Opening_Balance;
   ELSE Asset_MVNT = Assets - @PRIOR(Assets);
ENDIF;
```

This table shows the results when Essbase calculates the difference between the values of assets in successive months:

Assets -> Time	Opening_Balance	Jan	Feb	Mar
Assets	1200	1400	1300	1800
Asset_MVNT		**200**	**-100**	**500**

Essbase cycles through the months, performing these calculations:

1. The IF statement and @ISMBR function check whether the current member on the Year dimension is Jan. This check is necessary because the Asset_MVNT value for Jan cannot be calculated by subtracting the previous month's value.
2. If the current member on the Year dimension is Jan, Essbase subtracts the Opening_Balance from the Jan -> Assets value and places the result in Jan -> Asset_MVNT.
3. If the current member on the Year dimension is not Jan, the @PRIOR function obtains the value for the previous month's assets. Essbase subtracts the previous month's assets from the current month's assets. It places the result in the current month's Asset_MVNT value.

Testing for #MISSING Values

You can test for #MISSING values in a database. See "Consolidating #MISSING Values" on page 894.

Assume that a database outline contains a member called Commission. Commission is paid at 10% of sales when the Sales value for the current member combination is not #MISSING. When applied to a Commission member in the database outline, the following formula calculates Commission:

```
IF(Sales <> #MISSING) Commission = Sales * .1;
   ELSE Commission = #MISSING;
ENDIF;
```

If you place the formula in a calculation script, you must associate it with the Commission member as shown:

```
Commission(IF(Sales <> #MISSING) Commission = Sales * .1;
   ELSE Commission = #MISSING;
ENDIF;);
```

Essbase cycles through the database, performing the following calculations:

1. The IF statement checks the value of the Sales member for the current member combination.
2. If Sales is not #MISSING, Essbase multiplies the value in the Sales member by 0.1 and places the result in the Commission member.
3. If Sales is #MISSING, Essbase places #MISSING in the Commission member.

Calculating an Attribute Formula

You can perform specific calculations on attribute-dimension members in a database. See "Calculating Attribute Data" on page 174.

For example, to calculate profitability by ounce for products sized in ounces, you can use the @ATTRIBUTEVAL function in a calculation formula. In the Sample.Basic database, the Ratios branch of the Measures dimension contains a member called Profit per Ounce. The formula on this member:

```
Profit/@ATTRIBUTEVAL(@NAME(Ounces));
```

Essbase cycles through the Products dimension, performing the following calculations:

1. For each base member that is associated with a member from the Ounces attribute dimension, the @ATTRIBUTEVAL function returns the numeric attribute value (for example, 12 for the member 12 under Ounces).

   Note:

   The @NAME function is required to process the string "Ounces" before passing it to the @ATTRIBUTEVAL function.

2. Essbase then divides Profit by the result of @ATTRIBUTEVAL to yield Profit per Ounce.

Note:

See "Using Attributes in Calculation Formulas" on page 177. For more information about the @ATTRIBUTEVAL function, see the *Oracle Essbase Technical Reference*.

24 Defining Calculation Order

In This Chapter

The information in this chapter applies only to block storage databases and is not relevant to aggregate storage databases.

Also see:

- Chapter 57, "Comparison of Aggregate and Block Storage"
- "Sparse and Dense Dimensions" on page 66
- "Generations and Levels" on page 65
- "Understanding How Dynamic Calculation Changes Calculation Order" on page 425

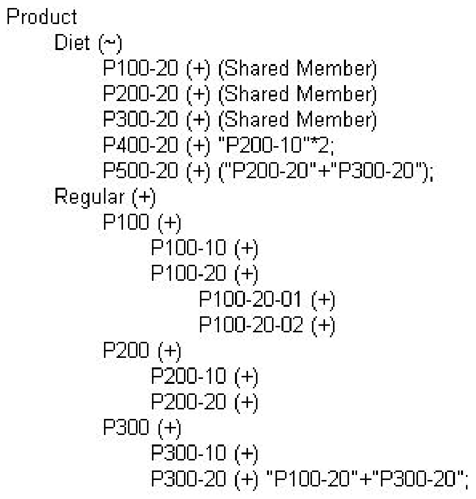
Data Storage in Data Blocks

Essbase stores data values in data blocks. Essbase creates a data block for each unique combination of sparse dimension members (providing that at least one data value exists for the combination).

Each data block contains all the dense dimension member values for its unique combination of sparse dimension members.

In the Sample.Basic database, the Year, Measures, and Scenario dimensions are dense; the Product and Market dimensions are sparse:

Figure 117 Dimensions from the Sample.Basic Database



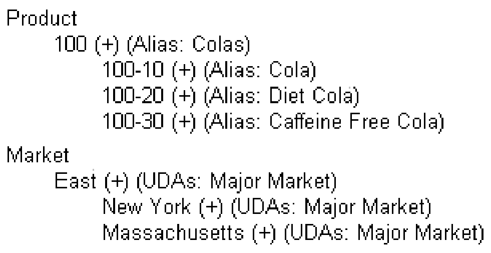
```
Product
    Diet (~)
        P100-20 (+) (Shared Member)
        P200-20 (+) (Shared Member)
        P300-20 (+) (Shared Member)
        P400-20 (+) "P200-10"*2;
        P500-20 (+) ("P200-20"+"P300-20");
    Regular (+)
        P100 (+)
            P100-10 (+)
            P100-20 (+)
                P100-20-01 (+)
                P100-20-02 (+)
        P200 (+)
            P200-10 (+)
            P200-20 (+)
        P300 (+)
            P300-10 (+)
            P300-20 (+) "P100-20"+"P300-20";
```

Note:

Sample.Basic also contains five attribute dimensions. These dimensions are sparse, Dynamic Calc, meaning that attribute data is not stored in the database. See Chapter 10, "Working with Attributes."

Essbase creates a data block for each unique combination of members in the Product and Market dimensions (providing that at least one data value exists for the combination). For example, it creates one data block for the combination of 100-10, New York. This data block contains all the Year, Measures, and Scenario values for 100-10, New York.

Figure 118 Product and Market Dimensions from the Sample.Basic Database



```
Product
    100 (+) (Alias: Colas)
        100-10 (+) (Alias: Cola)
        100-20 (+) (Alias: Diet Cola)
        100-30 (+) (Alias: Caffeine Free Cola)
Market
    East (+) (UDAs: Major Market)
        New York (+) (UDAs: Major Market)
        Massachusetts (+) (UDAs: Major Market)
```

In Essbase, member combinations are denoted by the cross-dimensional operator. The symbol for the cross-dimensional operator is -> (a hyphen followed by a greater-than symbol). So 100-10, New York is written 100-10 -> New York.

You can categorize data blocks in the following ways:

- Input

  These blocks are created by loading data to cells in a block. Input blocks can be created for (1) sparse, level 0 member combinations or (2) sparse, upper-level member combinations, when at least one of the sparse members is a parent-level member. Input blocks can be level 0 or upper-level blocks.

- Noninput

  These blocks are created through calculations. For example, in Sample.Basic, the East -> Cola block is created during a sparse calculation process (that is, the block did not exist before calculation).

- Level 0

  These blocks are created for sparse member combinations when all of the sparse members are level 0 members. For example, in Sample.Basic, New York -> Cola is a level 0 block because New York and Cola are level 0 members of their respective sparse dimensions. Level 0 blocks can be input or noninput blocks; for example, a level 0 noninput block is created during an allocation process, where data is loaded at a parent level and then allocated down to level 0.

- Upper level

  These blocks are created for sparse member combinations when at least one of the sparse members is a parent-level member. Upper-level blocks can be input or noninput blocks.

See "Generations and Levels" on page 65 and "Data Blocks and the Index System" on page 74.

Member Calculation Order

Essbase calculates a database at the data block level, bringing one or more blocks into memory and calculating the required values within the block. Essbase calculates the blocks in order, according to their block numbers. The database outline tells Essbase how to order the blocks. Within each block, Essbase calculates the values in order according to the hierarchy in the database outline. Therefore, overall, Essbase calculates a database based on the database outline.

When you perform a default calculation (CALC ALL) on a database, Essbase calculates the dimensions in this order:

- If both a dimension tagged as accounts and a dimension tagged as time exist, and if formulas are applied to members on the accounts dimension, Essbase calculates in this order:
  1. Dimension tagged as accounts
  2. Dimension tagged as time
  3. Other dense dimensions (in the order in which they are displayed in the database outline)
  4. Other sparse dimensions (in the order in which they are displayed in the database outline)
- Otherwise, Essbase calculates in this order:
  1. Dense dimensions (in the order in which they are displayed in the database outline)
  2. Sparse dimensions (in the order in which they are displayed in the database outline)

Note:

Attribute dimensions, which are not included in the database consolidation, do not affect calculation order. See Chapter 10, "Working with Attributes."

In the Sample.Basic database, the dimensions are calculated in this order: Measures, Year, Scenario, Product, and Market.

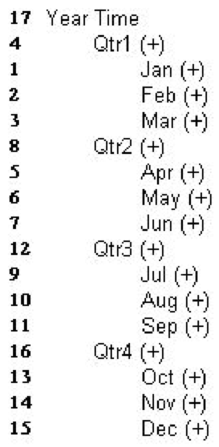
You can override the default order by using a calculation script. See Chapter 28, “Developing Calculation Scripts.”

Understanding the Effects of Member Relationships

The order of calculation within each dimension depends on the relationships between members in the database outline. Within each branch of a dimension, level 0 values are calculated first followed by their level 1, parent value. Then the level 0 values of the next branch are calculated, followed by their level 1, parent value. The calculation continues in this way until all levels are calculated.

Figure 119 shows the Year dimension from the Sample.Basic database. The calculation order is shown on the left. This example assumes that the parent members are not tagged as Dynamic Calc. See Chapter 26, “Dynamically Calculating Data Values.”

Figure 119 Year Dimension from the Sample.Basic Database



```
17  Year Time
4       Qtr1 (+)
1           Jan (+)
2           Feb (+)
3           Mar (+)
8       Qtr2 (+)
5           Apr (+)
6           May (+)
7           Jun (+)
12      Qtr3 (+)
9           Jul (+)
10          Aug (+)
11          Sep (+)
16      Qtr4 (+)
13          Oct (+)
14          Nov (+)
15          Dec (+)
```

Jan is the first member in the first branch. Jan has no formula, so it is not calculated. The same applies to Feb and Mar, the other two members in the branch.

Essbase calculates Qtr1 by consolidating Jan, Feb, and Mar. In this example, these members are added.

Essbase then calculates the Qtr2 through Qtr4 branches in the same way.

Finally, Essbase calculates the Year member by consolidating the values of Qtr1 through Qtr4. These members are added.

Determining Member Consolidation

You can choose how Essbase consolidates members by applying any calculation operator (+, -, /, *, %, ~, ^) to the members in the database outline.

If an accounts member has a time balance tag (First, Last, or Average), Essbase consolidates it accordingly. See "Calculating First, Last, and Average Values" on page 435.

If a parent member has a label only tag, Essbase does not calculate the parent from its children.

If a member has a ~ tag, Essbase does not consolidate the member up to its parent.

If a member has a ^ tag, Essbase does not consolidate the member in any dimension.

Note:

If you use dynamic calculations, Essbase may use a different calculation order. See "Calculation Order for Dynamic Calculation" on page 425.

Ordering Dimensions in the Database Outline

To ensure the required calculation results, consider the calculation order of the dimensions in the database outline if you do either of these tasks:

- Use calculation operators to divide (/), multiply (*), or calculate percentages (%) for members in the database outline.
- Place formulas on members in the database outline.

You need not consider calculation order if you use only calculation operators to add (+) and subtract (–) members in the database outline and you do not use formulas in the outline.

Placing Formulas on Members in the Database Outline

If you place formulas on members in the database outline, consider the calculation order of the dimensions. A formula that is attached to a member on one dimension may be overwritten by a subsequent calculation on another dimension.

For example, the Sample.Basic database has a Measures dimension, tagged as accounts, and a Year dimension, tagged as time. Measures is calculated first and Year second. If you attach a formula to Margin on the Measures dimension, Essbase calculates the formula when it calculates the Measures dimension. Essbase then overwrites the formula when it consolidates the Year dimension. See "Cell Calculation Order" on page 394.

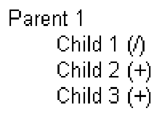
Using the Calculation Operators *, /, and %

If you use calculation operators to multiply (*), divide (/), and calculate percentages (%) for members in the database outline, consider the calculation order of the dimensions. The required calculated values may be overwritten by a subsequent calculation on another dimension.

For example, the Sample.Basic database has a Measures dimension, tagged as accounts, and a Year dimension, tagged as time. Measures is calculated first and Year second. If you multiply members on the Measures dimension, the calculated results may be overwritten when Essbase consolidates values on the Year dimension. See "Cell Calculation Order" on page 394.

When you use a multiplication (*), division (/), or percentage (%) operator to consolidate members, carefully order the members in the branch to achieve the required result.

Figure 120 Calculation Operators in the Database Outline



```
Parent 1
    Child 1 (/)
    Child 2 (+)
    Child 3 (+)
```

In Figure 120 on page 390, assume that the user wants to divide the total of Child 2 and Child 3 by Child 1. However, if Child 1 is the first member, Essbase starts with Child 1, starting with the value #MISSING, and dividing it by Child 1. The result is #MISSING. Essbase then adds Child 2 and Child 3. Obviously, this result is not the required one.

To calculate the correct result, make Child 1 the last member in the branch.

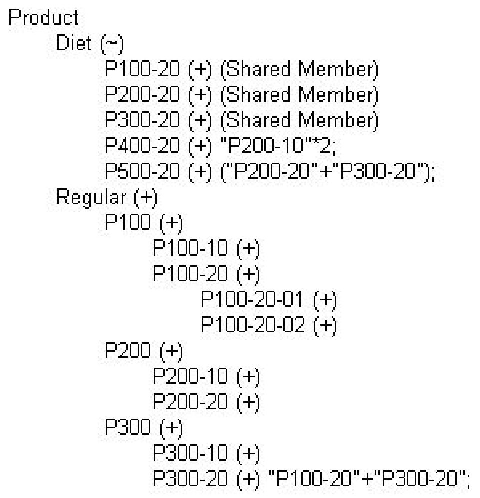
You can apply a formula to a member on the database outline to achieve the same result. However, it is far more efficient to use these calculation operators on members as shown in Figure 120 on page 390.

Avoiding Forward Calculation References

To obtain the calculation results you expect, ensure that the outline does not contain forward calculation references. *Forward calculation references* occur when the value of a calculating member is dependent on a member that Essbase has not yet calculated. In these cases, Essbase may not produce the required calculation results.

For example, consider this Product dimension:

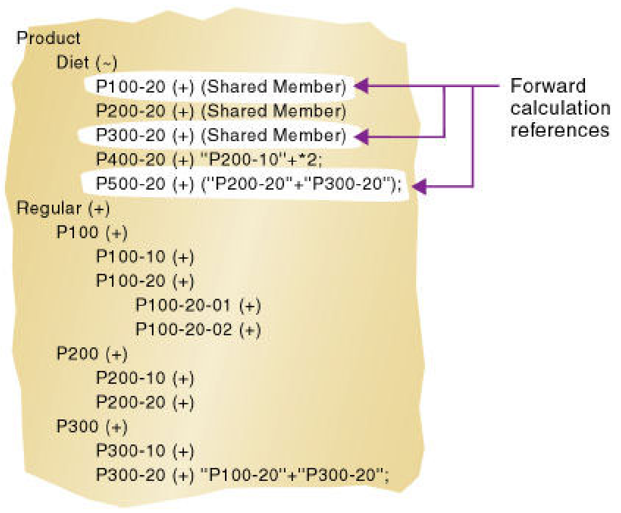
Figure 121 Example Product Dimension



```
Product
    Diet (~)
        P100-20 (+) (Shared Member)
        P200-20 (+) (Shared Member)
        P300-20 (+) (Shared Member)
        P400-20 (+) "P200-10"*2;
        P500-20 (+) ("P200-20"+"P300-20");
    Regular (+)
        P100 (+)
            P100-10 (+)
            P100-20 (+)
                P100-20-01 (+)
                P100-20-02 (+)
        P200 (+)
            P200-10 (+)
            P200-20 (+)
        P300 (+)
            P300-10 (+)
            P300-20 (+) "P100-20"+"P300-20";
```

This Product dimension has three forward calculation references. Two shared members and one nonshared member have forward calculation references, as shown in Figure 122 on page 391:

Figure 122 Example Product Dimension Showing Forward Calculation References


Product
Diet (~)
P100-20 (+) (Shared Member)
P200-20 (+) (Shared Member)
P300-20 (+) (Shared Member)
P400-20 (+) "P200-10"+*2;
P500-20 (+) ("P200-20"+"P300-20");
Regular (+)
P100 (+)
P100-10 (+)
P100-20 (+)
P100-20-01 (+)
P100-20-02 (+)
P200 (+)
P200-10 (+)
P200-20 (+)
P300 (+)
P300-10 (+)
P300-20 (+) "P100-20"+"P300-20";
Forward calculation references

In Outline Editor, when you verify the outline, Essbase identifies shared members with forward calculation references. Verifying the outline does *not* identify nonshared members that have forward calculation references. You can save and use an outline containing forward calculation references.

➤ To verify the outline, see "Verifying Outlines" in the *Oracle Essbase Administration Services Online Help*.

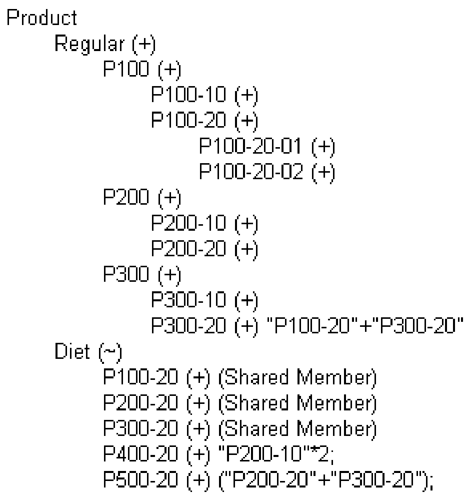
Consider the five members under Diet. The members P100-20, P300-20, and P500-20 have forward calculation references:

- P100-20 (+) (Shared Member): Essbase calculates the shared member P100-20 before it calculates the actual member P100-20. Because the actual member P100-20 has children, Essbase must calculate the actual member by adding its children before it can accurately calculate the shared member P100-20.
- P300-20 (+) (Shared Member): Essbase calculates the shared member P300-20 before it calculates the actual member P300-20. Because the actual member P300-20 has a formula, Essbase must calculate the actual member before it can accurately calculate the shared member P300-20.
- P500-20 (+) ("P200-20" + "P300-20"): The formula applied to P500-20 references members that Essbase has not yet calculated. One referenced member, P300-20, has its own formula, and Essbase must calculate P300-20 before it can accurately calculate P500-20. The members P200-20 and P400-20 calculate correctly, because they do not have forward calculation references.
- P200-20 (+) (Shared Member): P200-20 is *not* a forward calculation reference, although Essbase calculates the shared member P200-20 before it calculates the actual member P200-20. The actual member P200-20 has no calculation dependencies (no children and no formula). Therefore, Essbase does not need to calculate the actual member before the shared member. Essbase simply takes the value of the actual member.
- P400-20 (+) ("P200-10" * 2): P400-20 is *not* a forward calculation reference, although the formula that is applied to P400-20 references a member that Essbase has not yet calculated.

The member referenced in the formula does not itself have calculation dependencies. P200-10 is the only member in the formula, and P200-10 does not itself have children or a formula. Essbase accurately calculates P400-20.

To get accurate calculation results for P100-20, P300-20, and P500-20, change the order of members in the outline. By placing the Diet shared members after the Regular members, you ensure that Essbase calculates the members in the required order.

Figure 123 Changed Product Dimension Without Forward Calculation References



```
Product
    Regular (+)
        P100 (+)
            P100-10 (+)
            P100-20 (+)
                P100-20-01 (+)
                P100-20-02 (+)
        P200 (+)
            P200-10 (+)
            P200-20 (+)
        P300 (+)
            P300-10 (+)
            P300-20 (+) "P100-20"+"P300-20"
    Diet (~)
        P100-20 (+) (Shared Member)
        P200-20 (+) (Shared Member)
        P300-20 (+) (Shared Member)
        P400-20 (+) "P200-10"*2;
        P500-20 (+) ("P200-20"+"P300-20");
```

Now Essbase calculates:

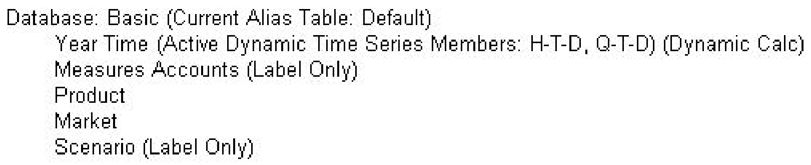
- The actual member P100-20 before it calculates the shared member P100-20. So, P100-20 no longer has a forward calculation reference.
- The actual member P300-20 before the shared member P300-20. So, P300-20 no longer has a forward calculation reference.
- The referenced member with a formula, P300-20, before the member P500-20. So, P500-20 no longer has a forward calculation reference.

Block Calculation Order

Essbase calculates blocks in the order in which the blocks are numbered. Essbase takes the first sparse dimension in a database outline as a starting point. It defines the sparse member combinations from this first dimension.

In the Sample.Basic database, Product is the first sparse dimension in the database outline.

Figure 124 Dimensions in the Sample.Basic Database



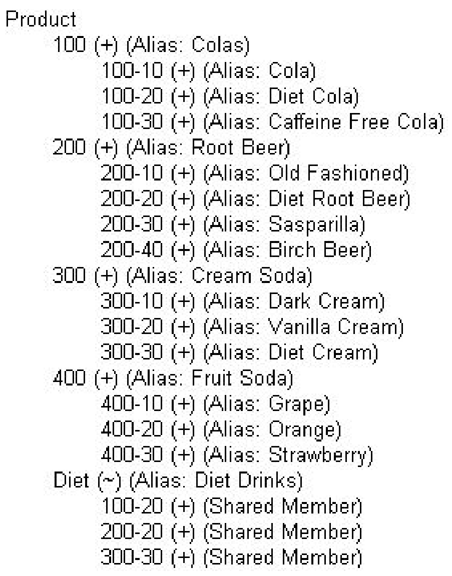
```
Database: Basic (Current Alias Table: Default)
    Year Time (Active Dynamic Time Series Members: H-T-D, Q-T-D) (Dynamic Calc)
    Measures Accounts (Label Only)
    Product
    Market
    Scenario (Label Only)
```

Note:

The attribute dimensions in the Sample.Basic outline (not shown in the figure above), are not included in the database consolidation and do not affect block calculation order. See Chapter 10, “Working with Attributes.”.

Product has 19 members (excluding the shared members, for which Essbase does not create data blocks). Therefore, the first 19 data blocks in the database are numbered according to the calculation order of members in the Product dimension.

Figure 125 Product Dimension from the Sample.Basic Database



```
Product
    100 (+) (Alias: Colas)
        100-10 (+) (Alias: Cola)
        100-20 (+) (Alias: Diet Cola)
        100-30 (+) (Alias: Caffeine Free Cola)
    200 (+) (Alias: Root Beer)
        200-10 (+) (Alias: Old Fashioned)
        200-20 (+) (Alias: Diet Root Beer)
        200-30 (+) (Alias: Sasparilla)
        200-40 (+) (Alias: Birch Beer)
    300 (+) (Alias: Cream Soda)
        300-10 (+) (Alias: Dark Cream)
        300-20 (+) (Alias: Vanilla Cream)
        300-30 (+) (Alias: Diet Cream)
    400 (+) (Alias: Fruit Soda)
        400-10 (+) (Alias: Grape)
        400-20 (+) (Alias: Orange)
        400-30 (+) (Alias: Strawberry)
    Diet (~) (Alias: Diet Drinks)
        100-20 (+) (Shared Member)
        200-20 (+) (Shared Member)
        300-30 (+) (Shared Member)
```

The other sparse dimension is Market. The first 19 data blocks contain the first member to be calculated in the Market dimension, which is New York.

The following table shows the sparse member combinations for the first five of these 19 data blocks:

Block Number	Product Member	Market Member
0	Cola (100-10)	New York
1	Diet Cola (100-20)	New York
2	Caffeine Free Cola (100-30)	New York
3	Colas (100)	New York
4	Old Fashioned (200-10)	New York

The next member in the Market dimension is Massachusetts. Essbase creates the next 19 data blocks for sparse combinations of each Product member and Massachusetts.

The following table shows the sparse member combinations for the block numbers 19 through 23:

Block Number	Product Member	Market Member
19	Cola (100-10)	Massachusetts
20	Diet Cola (100-20)	Massachusetts
21	Caffeine Free Cola (100-30)	Massachusetts
22	Colas (100)	Massachusetts
23	Old Fashioned (200-10)	Massachusetts

Essbase continues until blocks have been created for all combinations of sparse dimension members for which at least one data value exists.

Essbase creates a data block only if at least one value exists for the block. For example, if no data values exist for Old Fashioned Root Beer (200-10) in Massachusetts, then Essbase does not create a data block for 200-10 -> Massachusetts. However, Essbase does reserve the appropriate block number for 200-10 -> Massachusetts in case data is loaded for that member combination in the future.

When you run a default calculation (CALC ALL) on a database, each block is processed in order, according to its block number. If you have Intelligent Calculation turned on, and if the block does not need to be calculated, then Essbase skips the block and moves to the next block. For information about how intelligent calculation is used to optimize performance, see Chapter 25, "Understanding Intelligent Calculation."

Data Block Renumbering

Essbase renumbers the data blocks when you make any of these changes:

- Move a sparse dimension
- Add a sparse dimension
- Change a dense dimension to a sparse dimension
- Move any member in a sparse dimension
- Delete any member in a sparse dimension
- Add a member to a sparse dimension

Cell Calculation Order

Each data block contains all the dense dimension member values for its unique combination of sparse dimension members. Each data value is contained in a cell of the data block.

The order in which Essbase calculates the cells within each block depends on how you have configured the database. How you have configured the database defines the member calculation order of dense dimension members *within each block*. It also defines the calculation order of blocks that represent sparse dimension members.

Use the following sections to better understand cell calculation order.

Cell Calculation Order: Example 1

Consider the simplest case, in which both of these conditions are true:

- No dimensions have time or accounts tags.
- The setting for consolidating #MISSING values is turned on.

Market and Year are both dense dimensions. The following table shows a subset of the cells in a data block. Data values have been loaded into the input cells. Essbase calculates the shaded cells. The numbers in bold show the calculation order for these cells. The cell with multiple consolidation paths is darkly shaded.

Year -> Market	New York	Massachusetts	East
Jan	112345	68754	**3**
Feb	135788	75643	**4**
Mar	112234	93456	**5**
Qtr1	**1**	**2**	**6**

Essbase calculates dense dimensions in the order in which they display in the database outline. Assuming that the Year dimension is displayed before the Market dimension in the database outline, the Year dimension is calculated before the Market dimension.

The cells are calculated in this order:

1. Qtr1 -> New York
2. Qtr1 -> Massachusetts
3. Jan -> East
4. Feb -> East
5. Mar -> East
6. Qtr1 -> East

Qtr1 -> East has multiple consolidation paths. It can be consolidated on Market or on Year. When consolidated on Market, it is a consolidation of Qtr1 -> New York and Qtr1 -> Massachusetts. When consolidated on Year, it is a consolidation of Jan -> East, Feb -> East, and Mar -> East.

Essbase knows that Qtr1 -> East has multiple consolidation paths. Therefore, it calculates Qtr1 -> East only once and uses the consolidation path of the dimension calculated last. In the above example, this dimension is Market.

The results are shown in this table:

Year -> Market	New York	Massachusetts	East
Jan	112345	68754	**181099**
Feb	135788	75643	**211431**
Mar	112234	93456	**205690**
Qtr1	**360367**	**237853**	**598220**

Note:

Qtr1 -> East has been calculated only once by consolidating the values for Qtr1.

From the calculation order, you can see that if you place a member formula on Qtr1 in the database outline, Essbase ignores it when calculating Qtr1 -> East. If you place a member formula on East in the database outline, the formula is calculated when Essbase consolidates Qtr1 -> East on the Market consolidation path. If required, you can use a calculation script to calculate the dimensions in the order you choose. See Chapter 28, "Developing Calculation Scripts."

Cell Calculation Order: Example 2

Consider a second case, in which both of these conditions are true:

- No dimensions have time or accounts tags.
- The setting for consolidating #MISSING values is turned off (the default).

Market and Year are both dense dimensions. The following table shows a subset of the cells in a data block. Data values have been loaded into the input cells. Essbase calculates the shaded cells. The numbers in bold show the calculation order for these cells. The cell with multiple consolidation paths is darkly shaded.

Year -> Market	New York	Massachusetts	East
Jan	112345	68754	**4**
Feb	135788	75643	**5**
Mar	112234	93456	**6**
Qtr1	**1**	**2**	**3/7**

Essbase calculates dense dimensions in the order in which they are defined in the database outline. Assuming that the Year dimension is positioned before the Market dimension in the database outline, the Year dimension is calculated before the Market dimension.

The cells are calculated in this order:

1. Qtr1 -> New York
2. Qtr1 -> Massachusetts
3. Qtr1 -> East
4. Jan -> East
5. Feb -> East
6. Mar -> East
7. Qtr1 -> East

Qtr1 -> East is calculated on both the Year and Market consolidation paths. First, it is calculated as a consolidation of Qtr1 -> New York and Qtr1 -> Massachusetts. Second, it is calculated as a consolidation of Jan -> East, Feb -> East, and Mar -> East.

The results are identical to the previous case. However, Qtr1 -> East has been calculated twice. This fact is significant when you need to load data at parent levels (see "Cell Calculation Order: Example 3" on page 397).

Year -> Market	New York	Massachusetts	East
Jan	112345	68754	181099
Feb	135788	75643	211431
Mar	112234	93456	205690
Qtr1	360367	237853	598220

From the calculation order, you can see that if you place a member formula on Qtr1 in the database outline, its result is overwritten when Essbase consolidates Qtr1 -> East on the Market consolidation path. If you place a member formula on East in the database outline, the result is retained, because the Market consolidation path is calculated last.

Cell Calculation Order: Example 3

Consider the previous example, and add a third condition:

- No dimensions have time or accounts tags.
- The setting for consolidating #MISSING values is turned off (the default).
- Data values have been loaded at parent levels.

Market and Year are both dense dimensions. The following table shows a subset of the cells in a data block. Data values have been loaded into cells at the parent level.

Essbase calculates dense dimensions in the order in which they are defined in the database outline. Assuming that the Year dimension is positioned before the Market dimension in the database outline, the Year dimension is calculated before the Market dimension.

Year -> Market	New York	Massachusetts	East
Jan	#MISSING	#MISSING	181099
Feb	#MISSING	#MISSING	211431
Mar	#MISSING	#MISSING	205690
Qtr1	#MISSING	#MISSING	

The cells are calculated in the same order as in Example 2. Qtr1 -> East is calculated on both the Year and Market consolidation paths.

Because the setting for consolidating #MISSING values is turned off, Essbase does not consolidate the #MISSING values. Thus, the data that is loaded at parent levels is not overwritten by the #MISSING values below it.

However, if any of the child data values are not #MISSING, these values are consolidated and overwrite the parent values. For example, if Jan -> New York contains 50000.00, this value overwrites the values loaded at parent levels.

Essbase first correctly calculates the Qtr1 -> East cell by consolidating Jan -> East, Feb -> East, and Mar -> East. Second, it calculates on the Market consolidation path. However, it does not consolidate the #MISSING values in Qtr1 -> New York and Qtr1 -> Massachusetts, so the value in Qtr1 -> East is not overwritten.

This table shows the results:

Year -> Market	New York	Massachusetts	East
Jan	#MISSING	#MISSING	181099
Feb	#MISSING	#MISSING	211431
Mar	#MISSING	#MISSING	205690
Qtr1	#MISSING	#MISSING	598220

Essbase must calculate the Qtr1 -> East cell twice to ensure that a value is calculated for the cell. If Qtr1 -> East is calculated according to only the last consolidation path, the result is #MISSING, which is not the required result.

Cell Calculation Order: Example 4

Consider a case in which all of these conditions are true:

- The Year dimension is tagged as time.
- The Measures dimension is tagged as accounts.
- The setting for consolidating #MISSING values is turned off (the default).

Figure 126 shows the Profit branch of the Measures dimension in the Sample.Basic database. This example assumes that Total Expenses is not a Dynamic Calc member. See Chapter 26, "Dynamically Calculating Data Values."

Figure 126 Profit Branch of the Measures Dimension in the Sample.Basic Database

```
Profit (+)
    Margin (+)
        Sales (+)
        COGS (-) (Expense Reporting)
    Total Expenses (-) (Expense Reporting)
        Marketing (+) (Expense Reporting)
        Payroll (+) (Expense Reporting)
        Misc (+) (Expense Reporting)
```

The following table shows a subset of the cells in a data block. Data values have been loaded into the input cells. Essbase calculates the shaded cells. The numbers in bold show the calculation order for these cells. Cells with multiple consolidation paths are darkly shaded.

The Marketing, Payroll, and Misc Expenses values have been loaded at the Qtr1, parent level.

Measures/Year	Jan	Feb	Mar	Qtr1
Sales	31538	32069	32213	**13**
COGS	14160	14307	14410	**14**
Margin	**1**	**4**	**7**	**10/15**
Marketing	#MISSING	#MISSING	#MISSING	15839
Payroll	#MISSING	#MISSING	#MISSING	12168
Misc	#MISSING	#MISSING	#MISSING	233
Total Expenses	**2**	**5**	**8**	**11/16**
Profit	**3**	**6**	**9**	**12/17**

Essbase calculates a dimension tagged as accounts first, followed by a dimension tagged as time. Therefore, in the above example, Measures is calculated before Year.

Three cells have multiple consolidation paths:

- Margin -> Qtr1
- Total Expenses -> Qtr1
- Profit -> Qtr1

Because the setting for consolidating #MISSING values is turned off, Essbase does not consolidate the #MISSING values. Thus, any data that is loaded at parent levels is not overwritten by the #MISSING values and Essbase calculates the three cells with multiple consolidation paths twice.

The results are shown in this table:

Measures/Year	Jan	Feb	Mar	Qtr1
Sales	31538	32069	32213	95820
COGS	14160	14307	14410	42877
Margin	17378	17762	17803	52943
Marketing	#MISSING	#MISSING	#MISSING	15839
Payroll	#MISSING	#MISSING	#MISSING	12168
Misc	#MISSING	#MISSING	#MISSING	233
Total Expenses				28240
Profit	17378	17762	17803	12/17

From the calculation order, you can see that if you place a member formula on, for example, Margin in the database outline, its result is overwritten by the consolidation on Qtr1.

Cell Calculation Order for Formulas on a Dense Dimension

The cell calculation order within a data block is not affected by formulas on members. When Essbase encounters a formula in a data block, it locks any other required data blocks, calculates the formula, and proceeds with the data block calculation.

When placing a formula on a dense dimension member, carefully consider the cell calculation order. As described in the examples above, the dimension calculated last overwrites previous cell calculations for cells with multiple consolidation paths. If required, you can use a calculation script to change the order in which the dimensions are calculated. See Chapter 28, “Developing Calculation Scripts” and Chapter 22, “Developing Formulas.”

Calculation Passes

Whenever possible, Essbase calculates a database in one calculation pass through the database. Thus, it reads each of the required data blocks into memory only once, performing all relevant calculations on the data block and saving it. However, in some situations, Essbase must perform more than one calculation pass through a database. On subsequent calculation passes, Essbase brings data blocks back into memory, performs further calculations on them, and saves them again.

When you perform a default, full calculation of a database (CALC ALL), Essbase attempts to calculate the database in one calculation pass. If you have dimensions that are tagged as accounts or time, Essbase may have to do more than one calculation pass through the database.

The following table shows the number of calculation passes Essbase performs if you have dimensions that are tagged as time or accounts, and you have at least one formula on the accounts dimension:

Dimension Tagged As:			
Accounts	**Time**	**Calculation Passes**	**During each calculation pass, Essbase calculates based on:**
Dense or Sparse	None	1	All dimensions
Dense	Dense	1	All dimensions
Dense	Sparse	2	Pass 1: Accounts and time dimensions Pass 2: Other dimensions
Sparse	Sparse	2	Pass 1: Accounts and time dimensions Pass 2: Other dimensions
Sparse	Dense	2	Pass 1: Accounts dimension Pass 2: Other dimensions

If you are using formulas that are tagged as Two-Pass, Essbase may need to do an *extra* calculation pass to calculate these formulas. See "Using Two-Pass Calculation" on page 883.

When you use a calculation script to calculate a database, the number of calculation passes Essbase needs to perform depends upon the calculation script. See "Calculation Passes" on page 399 and "Understanding Multiple-Pass Calculations" on page 412. Also see "Grouping Formulas and Calculations" on page 454.

If the isolation level is set for committed access, and multiple passes are required, Essbase writes data values at the end of each pass. Data retrievals that occur between passes can pick up intermediate values.

When you calculate a database, Essbase automatically displays the calculation order of the dimensions for each pass through the database and tells you how many times Essbase has cycled through the database during the calculation. Essbase displays this information in the ESSCMD window and in the application log.

➤ To display the application log, see "Viewing the Essbase Server and Application Logs" on page 742.

For each data block, Essbase decides whether to do a dense or a sparse calculation. The type of calculation it chooses depends on the type of values within the data block. When you run a default calculation (CALC ALL) on a database, each block is processed in order, according to its block number.

Essbase calculates the blocks using this procedure:

- If you have Intelligent Calculation turned on, and if the block does not need to be calculated (if it is marked as *clean*), Essbase skips the block and moves to the next block. See Chapter 25, "Understanding Intelligent Calculation."
- If the block needs recalculating, Essbase checks to see if the block is a level 0, an input, or an upper-level block. See "Data Storage in Data Blocks" on page 385.
- If the block is a level 0 block or an input block, Essbase performs a dense calculation on the block. Each cell in the block is calculated. See "Cell Calculation Order" on page 394.
- If the block is an upper-level block, Essbase either consolidates the values or performs a sparse calculation on the data block.

  The sparse member combination of each upper-level block contains at least one parent member. Essbase consolidates or calculates the block based on the parent member's dimension. For example, if the upper-level block is for Product -> Florida from the Sample.Basic database, then Essbase chooses the Product dimension.

  If the sparse member combination for the block has more than one parent member, Essbase chooses the last dimension in the calculation order that includes a parent member. For example, if the block is for Product -> East, and you perform a default calculation on the Sample.Basic database, Essbase chooses the Market dimension, which contains East. The Market dimension is last in the default calculation order because it is placed after the Product dimension in the database outline. See "Member Calculation Order" on page 387.

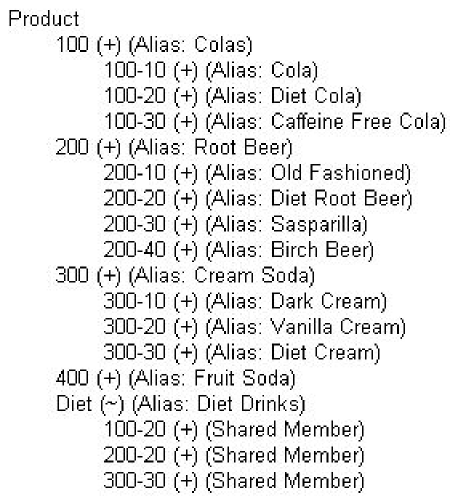
  Based on the chosen sparse dimension, Essbase either consolidates the values or performs a sparse calculation on the data block:

  - If a formula is applied to the data block member on the chosen sparse dimension, Essbase performs a formula calculation on the sparse dimension. Essbase evaluates each cell in the data block. The formula affects only the member on the sparse dimension, so overall calculation performance is not significantly affected.
  - If the chosen sparse dimension is a default consolidation, Essbase consolidates the values, taking the values of the previously calculated child data blocks.

Calculation of Shared Members

Shared members are those that share data values with other members. For example, in the Sample.Basic database, Diet Cola, Diet Root Beer, and Diet Cream are consolidated under two parents: under Diet and under their product types—Colas, Root Beer, and Cream Soda.

Figure 127 Calculating Shared Members



```
Product
    100 (+) (Alias: Colas)
        100-10 (+) (Alias: Cola)
        100-20 (+) (Alias: Diet Cola)
        100-30 (+) (Alias: Caffeine Free Cola)
    200 (+) (Alias: Root Beer)
        200-10 (+) (Alias: Old Fashioned)
        200-20 (+) (Alias: Diet Root Beer)
        200-30 (+) (Alias: Sasparilla)
        200-40 (+) (Alias: Birch Beer)
    300 (+) (Alias: Cream Soda)
        300-10 (+) (Alias: Dark Cream)
        300-20 (+) (Alias: Vanilla Cream)
        300-30 (+) (Alias: Diet Cream)
    400 (+) (Alias: Fruit Soda)
    Diet (~) (Alias: Diet Drinks)
        100-20 (+) (Shared Member)
        200-20 (+) (Shared Member)
        300-30 (+) (Shared Member)
```

The members under the Diet parent are shared members. See "Understanding Shared Members" on page 149.

A calculation on a shared member is a calculation on the actual member. If you use the FIX command to calculate a subset of a database and the subset includes a shared member, Essbase calculates the actual member.

25 Understanding Intelligent Calculation

In This Chapter

Introducing Intelligent Calculation

By default, when Essbase performs a full calculation of a database, it tracks which data blocks it calculates. If you then load a subset of data, on subsequent calculations, Essbase calculates only the data blocks that have not been calculated and the calculated blocks that require recalculation because of the new data. This process is called Intelligent Calculation.

By default, Intelligent Calculation is turned on. You can change this default setting in `essbase.cfg`. See the *Oracle Essbase Technical Reference*.

You can also turn Intelligent Calculation on or off in a calculation script. See "Turning Intelligent Calculation On and Off" on page 406.

For information on other calculation optimization methods, see:

- Chapter 14, "Designing Partitioned Applications"
- Chapter 26, "Dynamically Calculating Data Values"
- Chapter 55, "Optimizing Calculations"

Benefits of Intelligent Calculation

Intelligent Calculation is designed to provide significant calculation performance benefits for these types of calculations:

- A full calculation of a database (CALC ALL), with some exceptions.

  See "Limitations of Intelligent Calculation" on page 405.
- A calculation script that calculates all members in one CALC DIM statement.
- For database calculations that cannot use Intelligent Calculation for the full calculation, you may be able to use Intelligent Calculation for part of the calculation.

For example, to significantly improve calculation performance for a case in which you calculate a database by doing a default consolidation and then an allocation of data, enable Intelligent Calculation for the default consolidation and then disable Intelligent Calculation for the allocation.

Assuming that Intelligent Calculation is turned on (the default), create a calculation script to perform these steps for a partial Intelligent Calculation:

- ❍ Enable Intelligent Calculation, if it is disabled
- ❍ Use CALC ALL to calculate the database
- ❍ Use the SET UPDATECALC command to disable Intelligent Calculation
- ❍ Allocate data
- ❍ Optionally, enable Intelligent Calculation again

Intelligent Calculation and Data Block Status

To provide Intelligent Calculation, Essbase checks the status of the data blocks in a database. Data blocks have a calculation status of clean or dirty. Essbase marks a data block as clean after certain calculations.

When Intelligent Calculation is enabled, Essbase calculates only dirty blocks and their dependent parents. When disabled, Essbase calculates all data blocks, regardless of whether they are marked as clean or dirty.

Marking Blocks as Clean

Essbase marks data blocks as clean in these types of calculations:

- A full calculation (CALC ALL) of a database (the default calculation).
- A calculation script that calculates all the dimensions in one CALC DIM statement.

  For example, the following calculation script calculates all members in the Sample.Basic database:

  ```
  CALC DIM(Measures, Product, Market, Year, Scenario);
  ```

  Compare this calculation script to a calculation script that calculates all the members with two CALC DIM statements:

  ```
  CALC DIM(Measures, Product);
  CALC DIM(Market, Year, Scenario);
  ```

Using two CALC DIM statements causes Essbase to do at least two calculation passes through the database. In this calculation, Essbase does not, by default, mark the data blocks as clean. Because Intelligent Calculation depends on accurate clean and dirty status, you must manage these markers carefully. See “Maintaining Clean and Dirty Status” on page 405.

Essbase marks calculated data blocks as clean only in the situations described above, unless you use the SET CLEARUPDATESTATUS command in a calculation script. See “Using the SET CLEARUPDATESTATUS Command” on page 407.

Marking Blocks as Dirty

Essbase marks a data block as dirty in these situations:

- Calculating the data block for a partial calculation of the database only if SET CLEARUPDATESTATUS AFTER is not part of the partial calculation statement in the calculation script
- Loading data into the data block
- Restructuring the database (for example, by adding a member to a dense dimension)
- Copying data to the data block; for example, using DATACOPY

Maintaining Clean and Dirty Status

To use Intelligent Calculation when calculating a subset of a database or when performing multiple calculation passes through a database, consider carefully the implications of how Essbase marks data blocks as clean. When using Intelligent Calculation, you must accurately maintain the clean and dirty status of the data blocks to ensure that Essbase recalculates the database as efficiently as possible.

For example, when you calculate a subset of a database, the newly calculated data blocks are not marked as clean by default. You can ensure that the newly calculated blocks are marked as clean by using the SET CLEARUPDATESTATUS AFTER command in a calculation script. Before creating the calculation script, see "Using the SET CLEARUPDATESTATUS Command" on page 407 and the *Oracle Essbase Technical Reference*.

Limitations of Intelligent Calculation

Consider the following limitations and situations when using Intelligent Calculation:

- Intelligent Calculation works on a data block level and not on a cell level. For example, if you load a data value into one cell of a data block, the whole data block is marked as dirty.
- A CALC ALL that requires two passes through the database may calculate incorrectly. The problem occurs because blocks that are marked clean during the first pass are skipped during the second pass. To avoid this problem, turn Intelligent Calculation off or perform a CALC DIM for each dimension (rather than a CALC ALL for the database). A CALC ALL requires two passes through the database in either of these situations:
  - When the accounts dimension is sparse
  - When the accounts dimension is dense, the time dimension is sparse, and there is at least one more dense dimension in the outline
- Changing a formula on the database outline or changing an accounts property on the database outline does not cause Essbase to restructure the database. Therefore, Essbase does not mark the affected blocks as dirty. You must recalculate the appropriate data blocks. See "Changing Formulas and Accounts Properties" on page 415.
- Whenever possible, Essbase calculates formulas that are tagged as two-pass and in the dimension tagged as accounts as part of the main calculation of a database. You may,

however, need to use a calculation script to calculate some formulas twice. When you use a calculation script, disable Intelligent Calculation before recalculating formulas.

- When SET CREATENONMISSINGBLK is set to ON in a calculation script, Intelligent Calculation is turned off, and affected blocks are calculated whether they are marked clean or dirty.

Using Intelligent Calculation

This section provides information on turning Intelligent Calculation on and off and using Intelligent Calculation with different types of calculations.

Turning Intelligent Calculation On and Off

By default, Intelligent Calculation is turned on. To change the default, use the UPDATECALC setting in the `essbase.cfg` file.

To turn Intelligent Calculation on and off for the duration of a calculation script, use the SET UPDATECALC command in a calculation script.

See the *Oracle Essbase Technical Reference*.

Using Intelligent Calculation for a Default, Full Calculation

Intelligent Calculation provides significant performance benefits when you do a full calculation (CALC ALL) of a database. If you do a full calculation, leave Intelligent Calculation turned on (the default) to take advantage of its performance benefits.

➤ To check the current calculation setting, see “Setting the Default Calculation” in the *Oracle Essbase Administration Services Online Help*.

Caution!

When using Intelligent Calculation, note the information in “Limitations of Intelligent Calculation” on page 405.

Calculating for the First Time

When you do the first full calculation of a database, Essbase calculates every block. The performance is the same whether Intelligent Calculation is on or off.

Recalculating

When you do a full recalculation of a database with Intelligent Calculation turned on, Essbase checks each block to see whether it is marked as clean or dirty. See "Intelligent Calculation and Data Block Status" on page 404.

Checking data blocks has a 5% to 10% performance overhead, which is insignificant when compared to the performance gained by enabling Intelligent Calculation.

If, however, you recalculate a database in which more than approximately 80% of the values have changed, the overhead of Intelligent Calculation may outweigh the benefits. In this case, disable Intelligent Calculation.

Using Intelligent Calculation for a Calculation Script, Partial Calculation

Essbase marks a data block as clean when it calculates the data block on a full calculation (CALC ALL) or when it calculates all dimensions in one CALC DIM command. See "Marking Blocks as Clean" on page 404.

In any other calculations, Essbase does not mark calculated data blocks as clean, unless you use the SET CLEARUPDATESTATUS command in a calculation script. For example, if you calculate a subset of a database or calculate a database in two calculation passes, Essbase does not mark the calculated blocks as clean, unless you use the SET CLEARUPDATESTATUS command.

The following calculation scripts do not cause Essbase to mark the calculated data blocks as clean:

```
FIX("New York")
   CALC DIM(Product, Measures);
ENDFIX
CALC DIM(Measures, Product);
CALC DIM(Market, Year, Scenario);
```

Use SET CLEARUPDATESTATUS to avoid unnecessary recalculations.

Using the SET CLEARUPDATESTATUS Command

In some cases, Essbase does not mark calculated blocks as clean; for example, if you calculate a subset of a database or calculate a database in two calculation passes. To manually mark data blocks as clean for purposes of Intelligent Calculation, use the SET CLEARUPDATESTATUS command in a calculation script. Read this section, and also see "Intelligent Calculation and Data Block Status" on page 404.

Understanding SET CLEARUPDATESTATUS

The SET CLEARUPDATESTATUS command has three parameters—AFTER, ONLY, and OFF.

- SET CLEARUPDATESTATUS AFTER;

Essbase marks calculated data blocks as clean, even if it is calculating a subset of a database.

- SET CLEARUPDATESTATUS ONLY;

  Essbase marks the specified data blocks as clean but does not calculate the data blocks. This parameter provides the same result as AFTER, but without calculation.

- SET CLEARUPDATESTATUS OFF;

  Essbase calculates the data blocks but does not mark the calculated data blocks as clean. Data blocks are not marked as clean, even on a full calculation (CALC ALL) of a database. The existing clean or dirty status of the calculated data blocks remains unchanged.

Choosing a SET CLEARUPDATESTATUS Setting

When you use the SET CLEARUPDATESTATUS command to mark calculated data blocks as clean, be aware of these recommendations before selecting the parameter (AFTER, ONLY, OFF):

- Only calculated data blocks are marked as clean.
- Do not use the SET CLEARUPDATESTATUS AFTER command with concurrent calculations unless you are certain that the concurrent calculations do not need to calculate the same data block or blocks. If concurrent calculations attempt to calculate the same data blocks, with Intelligent Calculation enabled, Essbase does not recalculate the data blocks if the data blocks are already marked clean by the other concurrent calculation. See "Handling Concurrent Calculations" on page 411.
- When Essbase calculates data blocks on a first calculation pass through a database, it marks the data blocks as clean. If you try to calculate the same data blocks on a subsequent pass with Intelligent Calculation enabled, Essbase does not recalculate the data blocks, because they are already marked as clean.

Reviewing Examples That Use SET CLEARUPDATESTATUS

Assume a scenario using the Sample.Basic database:

- Sparse dimensions are Market and Product.
- New York is a member on the sparse Market dimension.
- Intelligent Calculation is turned on (the default).

These examples show different ways of using SET CLEARUPDATESTATUS:

Example 1: CLEARUPDATESTATUS AFTER

```
SET CLEARUPDATESTATUS AFTER;
FIX("New York")
   CALC DIM(Product);
ENDFIX
```

In this example, Essbase searches for dirty parent data blocks for New York (for example New York -> Colas, in which Colas is a parent member on the Product dimension). It calculates these dirty blocks and marks them as clean. Essbase does not mark the level 0 data blocks as clean, because they are not calculated. For information on level 0 blocks, see Chapter 24, “Defining Calculation Order.”

Example 2: CLEARUPDATESTATUS ONLY

```
SET CLEARUPDATESTATUS ONLY;
FIX(“New York”)
   CALC DIM(Product);
ENDFIX
```

Essbase searches for dirty parent data blocks for New York (for example New York -> Colas, in which Colas is a parent member on the Product dimension). Essbase marks the dirty parent data blocks as clean but does not calculate the data blocks. Essbase does not mark the level 0 data blocks as clean because they are not calculated. For example, if New York -> 100-10 (a level 0 block) is dirty, it remains dirty.

Example 3: CLEARUPDATESTATUS OFF

```
SET CLEARUPDATESTATUS OFF;
CALC ALL;
CALC TWOPASS;
SET CLEARUPDATESTATUS ONLY;
CALC ALL;
```

In this example, Essbase first calculates all the dirty data blocks in the database. The calculated data blocks remain dirty. Essbase does not mark them as clean.

Essbase then calculates the members tagged as two-pass that are in the dimension tagged as accounts. Because the data blocks are still marked as dirty, Essbase recalculates them. Again, it does not mark the calculated data blocks as clean.

Essbase then searches for all the dirty blocks in the database and marks them as clean. It does not calculate the blocks, although a CALC ALL command is used.

Calculating Data Blocks

Essbase creates a data block for each unique combination of sparse dimension members, provided that at least one data value exists for the combination. Each data block represents all dense dimension member values for that unique combination of sparse dimension members.

For example, in the Sample.Basic database, the Market and Product dimensions are sparse. Therefore, the data block New York -> Colas represents all the member values on the Year, Measures, and Scenario dimensions for the sparse combination New York -> Colas.

These sections assume that you are familiar with the concepts of upper-level, level 0, and input data blocks. See “Data Storage in Data Blocks” on page 385.

Calculating Dense Dimensions

When you calculate a dense dimension and do not use a FIX command, Essbase calculates at least some of the data values in every data block in the database.

For example, the following calculation script is based on the Sample.Basic database:

```
SET CLEARUPDATESTATUS AFTER;
CALC DIM(Year);
```

This script calculates the Year dimension, which is a dense dimension. Because Year is dense, every data block in the database includes members of the Year dimension. Therefore, Essbase calculates data values in every data block. Because the script uses the SET CLEARUPDATESTATUS AFTER command, Essbase marks all data blocks as clean.

Calculating Sparse Dimensions

When you calculate a sparse dimension, Essbase may not need to calculate every data block in the database.

For example, the following calculation script is based on the Sample.Basic database:

```
SET CLEARUPDATESTATUS AFTER;
CALC DIM(Product);
```

This script calculates the Product dimension, which is a sparse dimension. Because Product is sparse, a data block exists for each member on the Product dimension. For example, one data block exists for New York -> Colas and another for New York -> 100-10.

Level 0 Effects

The data block New York -> 100-10 is a level 0 block; it does not represent a parent member on either sparse dimension (Market or Product). The data values for New York -> 100-10 are input values; they are loaded into the database. Therefore, Essbase does not need to calculate this data block. Nor does Essbase mark the data block for New York -> 100-10 as clean, even though the script uses the SET CLEARUPDATESTATUS AFTER command.

Note:

Essbase calculates level 0 data blocks if a corresponding sparse, level 0 member has a formula applied to it.

If you load data into a database, the level 0 data blocks into which you load data are marked as dirty. If you subsequently calculate only a sparse dimension or dimensions, the level 0 blocks remain dirty, because Essbase does not calculate them. Therefore, when you recalculate only a sparse dimension or dimensions, Essbase recalculates all upper-level data blocks, because the upper-level blocks are marked as dirty if their child blocks are dirty, although the upper-level blocks were originally clean.

Upper-Level Effects

Colas is a parent-level member on the Product dimension. Essbase must calculate values for Colas, so Essbase calculates this data block. Because the script uses the SET CLEARUPDATESTATUS AFTER command, Essbase marks the data block as clean.

When Essbase calculates a sparse dimension, it recalculates an upper-level data block if the block is dependent on one or more dirty child blocks.

Unnecessary Calculation

You can avoid unnecessary calculation by calculating at least one dense dimension. When you calculate a dense dimension and do not use the FIX command, data values are calculated in every data block, including the level 0 blocks. So the level 0 blocks are marked as clean.

Handling Concurrent Calculations

If concurrent calculations attempt to calculate the same data blocks, and Intelligent Calculation is turned on, Essbase may not recalculate the data blocks, because they are already marked as clean.

In the following example, based on the Sample.Basic database, Actual and Budget are members of the dense Scenario dimension. Because Scenario is dense, each data block in the database contains Actual and Budget values. If User 1 runs the following calculation script, Essbase calculates the Actual values for all data blocks that represent New York. Essbase marks the calculated data blocks as clean, although not all the data values in each calculated block have been calculated. For example, the Budget values have not been calculated.

```
SET CLEARUPDATESTATUS AFTER;
FIX("New York", Actual)
   CALC DIM(Product, Year);
ENDFIX
```

If User 2 runs the following calculation script to calculate the Budget values for New York, Essbase does not recalculate the specified data blocks, because they are already marked as clean. The calculation results for Budget are not correct.

```
SET CLEARUPDATESTATUS AFTER;
FIX("New York", Budget)
   CALC DIM(Product, Year);
ENDFIX
```

One way to solve this problem is to make the Scenario dimension sparse. Then the Actual and Budget values are in different data blocks; for example, New York -> Colas -> Actual and New York -> Colas -> Budget. In this case, the second calculation script correctly calculates Budget data block.

Running concurrent calculations might require an increase in the data cache. See "Sizing the Data Cache" on page 829.

Understanding Multiple-Pass Calculations

Whenever possible, Essbase calculates a database in one calculation pass through the database. See “Calculation Passes” on page 399.

When you use a calculation script to calculate a database, the number of calculation passes that Essbase performs depends upon the calculation script. See “Intelligent Calculation and Data Block Status” on page 404 and “Grouping Formulas and Calculations” on page 454.

For example, assume that Essbase calculates data blocks on a first calculation pass through a database and marks them as clean. If you attempt to calculate the same data blocks on a subsequent pass and Intelligent Calculation is enabled, Essbase does not recalculate the data blocks, because they are already marked as clean.

Reviewing Examples and Solutions for Multiple-Pass Calculations

These examples describe situations that produce incorrect calculation results and provide a solution to obtain correct results. They are based on the Sample.Basic database and assume that Intelligent Calculation is turned on.

Example 1: Intelligent Calculation and Two-Pass

This calculation script does a default calculation and then a two-pass calculation:

```
CALC ALL;
CALC TWOPASS;
```

Error

Essbase calculates the dirty data blocks in the database and marks all the data blocks as clean. Essbase then needs to recalculate the members tagged as two-pass in the dimension tagged as accounts. However, Essbase does not recalculate the specified data blocks because they are already marked as clean. The calculation results are not correct.

Solution

You can calculate the correct results by disabling Intelligent Calculation for the two-pass calculation.

Example 2: SET CLEARUPDATESTATUS and FIX

This calculation script calculates data values for New York. The calculation is based on the Product dimension:

```
SET CLEARUPDATESTATUS AFTER;
FIX(“New York”)
   CALC DIM(Product);
ENDFIX
CALC TWOPASS;
```

Error

➤ Essbase performs the following processes:

1. Essbase cycles through the database calculating the dirty data blocks that represent New York. The calculation is based on the Product dimension. Thus, Essbase calculates only the blocks that represent a parent member on the Product dimension (for example, New York -> Colas, New York -> Root Beer, and New York -> Fruit Soda), and then only calculates the aggregations and formulas for the Product dimension.
2. Because the SET CLEARUPDATESTATUS AFTER command is used, Essbase marks the calculated data blocks as clean, although not all data values in each calculated block have been calculated.
3. Essbase should recalculate the members tagged as two-pass in the dimension tagged as accounts; however, some of these data blocks are already marked as clean from the calculation in step 2. Essbase does not recalculate the data blocks that are marked as clean. The calculation results are not correct.

Solution

You can calculate the correct results by disabling Intelligent Calculation for the two-pass calculation.

Example 3: SET CLEARUPDATESTATUS and Two CALC DIM Commands

This calculation script bases the database calculation on the Product and Year dimensions. Because two CALC DIM commands are used, Essbase does two calculation passes through the database:

```
SET CLEARUPDATESTATUS AFTER;
CALC DIM(Product);
CALC DIM(Year);
```

Error

➤ Essbase performs the following processes:

1. Essbase cycles through the database calculating the dirty data blocks. The calculation is based on the Product dimension, as in "Example 2: SET CLEARUPDATESTATUS and FIX" on page 412.
2. Because the SET CLEARUPDATESTATUS AFTER command is used, Essbase marks the calculated data blocks as clean, although not all data values in each calculated block have been calculated.
3. Essbase should recalculate the data blocks. The recalculation is based on the Year dimension. However, as a result of the calculation in step 2, some data blocks are already marked as clean, and Essbase does not recalculate them. The calculation results are not correct.

Solution

You can calculate the correct results by using one CALC DIM command to calculate the Product and Year dimensions. Essbase calculates both dimensions in one calculation pass through the database. The following calculation script calculates the correct results:

```
SET CLEARUPDATESTATUS AFTER;
CALC DIM(Product, Year);
```

Note:

When you calculate several dimensions in one CALC DIM command, Essbase calculates the dimensions in the default calculation order and not in the order in which you list them in the command. See "Member Calculation Order" on page 387.

Example 4: Two Calculation Scripts

This example calculates data values for New York but calculates based on two dimensions using two calculation scripts. The first calculation script calculates the Product dimension:

```
SET CLEARUPDATESTATUS AFTER;
FIX("New York")
   CALC DIM(Product);
ENDFIX
```

Essbase calculates the data blocks that include New York. Because the calculation is based on the Product dimension, Essbase calculates only the dirty blocks that include a parent member on the Product dimension (for example, New York -> Colas, New York -> Root Beer, and New York -> Fruit Soda), and calculates only the aggregations and formulas for the Product dimension.

Because of the CLEARUPDATESTATUS AFTER command, Essbase marks the calculated data blocks as clean, although not all data values in each calculated block have been calculated.

The second calculation script calculates the Year dimension:

```
SET CLEARUPDATESTATUS AFTER;
FIX("New York")
   CALC DIM(Year);
ENDFIX
```

Essbase calculates the data blocks that represent New York. Because the calculation is based on the Year dimension, which is a dense dimension, Essbase should calculate all data blocks that include New York, although within each block Essbase calculates only the aggregations and formulas for the Year dimension.

Error

As a result of the first calculation, some data blocks for New York are already marked as clean. Essbase does not recalculate these data blocks with the second calculation script because the data blocks are marked as clean. The calculation results are not correct.

Solution

You can calculate the correct results by telling Essbase not to mark the calculated data blocks as clean. The following calculation script calculates the correct results:

```
SET CLEARUPDATESTATUS OFF;
FIX("New York")
   CALC DIM(Product);
ENDFIX
SET CLEARUPDATESTATUS AFTER;
FIX("New York")
```

```
   CALC DIM(Year);
ENDFIX
```

With the SET CLEARUPDATESTATUS OFF command, Essbase calculates dirty data blocks but does not to mark them as clean, unlike the SET CLEARUPDATESTATUS AFTER command.

This solution assumes that the data blocks are not marked as clean from a previous partial calculation of the database.

You can ensure that all data blocks are calculated, regardless of their status, by disabling Intelligent Calculation. The following calculation script calculates all specified data blocks, regardless of their clean or dirty status:

```
SET UPDATECALC OFF;
FIX("New York")
   CALC DIM(Year, Product);
ENDFIX
```

Because you have not used the SET CLEARUPDATESTATUS AFTER command, Essbase does not mark calculated data blocks as clean.

Understanding the Effects of Intelligent Calculation

Using Intelligent Calculation may have implications for how you administer a database. This section discusses the implications of each action.

Changing Formulas and Accounts Properties

Because neither changing a formula in the database outline nor changing an accounts property in the database outline causes Essbase to restructure the database, data blocks affected by such a change are not marked as dirty. For example, if you change a time balance tag in the dimension tagged as accounts, Essbase does not restructure the database and does not mark the affected blocks as dirty.

When you subsequently run a default calculation with Intelligent Calculation turned on, the changes are not calculated. To recalculate the appropriate data blocks, use a calculation script to perform any of the following tasks:

- Disable Intelligent Calculation and calculate the member formula that has changed.
- Disable Intelligent Calculation and use the FIX command to calculate the appropriate subset of a database.
- Disable Intelligent Calculation and perform a default CALC ALL on a database.

Using Relationship and Financial Functions

If you use relationship functions (for example, @PRIOR or @NEXT) or financial functions (for example, @ACCUM, @NPV, or @INTEREST) in a formula on a sparse dimension or a dense dimension, Essbase always recalculates the data block that contains the formula.

See the *Oracle Essbase Technical Reference*.

Restructuring Databases

When you restructure a database (for example, by adding a member to a dense dimension), all data blocks potentially need recalculating. Therefore, Essbase marks all data blocks as dirty. When you calculate the restructured database, all blocks are calculated.

Note:

Changing a formula in the database outline or changing an accounts property in the database outline does not cause Essbase to restructure the database. You must recalculate the appropriate data blocks. See "Changing Formulas and Accounts Properties" on page 415.

Copying and Clearing Data

When you copy values to a data block by using the DATACOPY command, the resulting data block is marked as dirty. Essbase calculates the block when you recalculate a database.

When you clear data values by using the CLEARDATA and CLEARBLOCK commands, Essbase clears all the blocks regardless of how they are marked.

Converting Currencies

When you convert currencies using the CCONV command, the resulting data blocks are marked as dirty. Essbase calculates all converted blocks when you recalculate a database.

26 Dynamically Calculating Data Values

In This Chapter

The information in this chapter applies only to block storage databases and is not relevant to aggregate storage databases. Also see Chapter 57, "Comparison of Aggregate and Block Storage."

Understanding Dynamic Calculation

When you design the overall database calculation, it may be more efficient to calculate some member combinations when you retrieve their data, instead of precalculating the member combinations during a batch database calculation. Dynamically calculating some values in a database can significantly improve the performance of an overall database calculation.

In Essbase, you can define a member to have a *dynamic calculation*. This definition tells Essbase to calculate a data value for the member as users request it. Dynamic calculation shortens batch database calculation time, but may increase retrieval time for the dynamically calculated data values. See "Reducing the Impact on Retrieval Time" on page 428.

In Essbase you specify dynamic calculations on a per-member basis. You can define a member in the database outline as one of two types of a dynamically calculated member:

- Dynamic Calc
- Dynamic Calc and Store

Understanding Dynamic Calc Members

For a member tagged as Dynamic Calc, Essbase does not calculate its data value during a batch database calculation (for example, during a CALC ALL). Instead, Essbase calculates the data value upon retrieval (for example, when you retrieve the data into Spreadsheet Add-in or Smart View.)

Specifically, Essbase calculates a data value dynamically when you request the data value in either of two ways:

- By retrieving the data value into Spreadsheet Add-in or Smart View
- By running a report script that displays the data value

Essbase does not store the calculated value; it recalculates the value for each subsequent retrieval.

Understanding Dynamic Calc and Store Members

Essbase calculates the data value for a member tagged as Dynamic Calc and Store when you retrieve the data, in the same way as for a Dynamic Calc member. For a Dynamic Calc and Store member, however, Essbase stores the data value that is calculated dynamically. Subsequent retrievals of that data value do not require recalculation, unless Essbase detects that the value needs recalculating.

Recalculation of Data

When Essbase detects that the data value for a Dynamic Calc and Store member needs recalculating, it places an indicator on the data block that contains the value, so that Essbase knows to recalculate the block on the next retrieval of the data value.

Essbase places the indicator on the data block containing the value and not on the data value itself, meaning that Essbase tracks Dynamic Calc and Store members at the data block level. See "Data Blocks and the Index System" on page 74.

If the data block needs recalculating, Essbase detects the need and places an indicator on the data block when any of the following situations occur:

- You perform a batch calculation.
- You restructure the database.
- You use the CLEARBLOCK DYNAMIC calculation command.

  See the *Oracle Essbase Technical Reference*.

Essbase recalculates the indicated data blocks when you next retrieve the data value.

Effect of Updated Values on Recalculation

Because Essbase does not detect that a data block needs recalculating and does not place an indicator on the data block when you update the data, updated blocks are recalculated only during the next batch calculation. Consider these scenarios:

- You do a data load.
- You do a Lock and Send from Spreadsheet Add-in.

If you load data into the children of a Dynamic Calc and Store member, and the member is a consolidation of its child members, Essbase does not know to recalculate the Dynamic Calc and Store member during the next retrieval. The parent member is recalculated only during the next batch calculation.

After loading data, you must perform a batch calculation of the database or use the CLEARBLOCK DYNAMIC calculation command to ensure that the Dynamic Calc and Store members are recalculated. See the *Oracle Essbase Technical Reference*.

Retrieving the Parent Value of Dynamically Calculated Child Values

If you retrieve a parent value that is calculated from Dynamic Calc or Dynamic Calc and Store child members, Essbase must dynamically calculate the child member combinations before calculating the parent value. Essbase does not store the child values, even if they are Dynamic Calc and Store members.

For example, assume that Market is a parent member and that East and West are Dynamic Calc and Store child members that consolidate up to Market. When you retrieve a data value for Market, Essbase calculates East and West, even though you have not specifically retrieved them. However, Essbase does not store the values of East or West.

Benefitting from Dynamic Calculation

Dynamically calculating some database values can significantly improve the performance of an overall database calculation.

By calculating some data values dynamically, you reduce:

- Batch calculation time of the database, because Essbase has fewer member combinations to calculate.
- Disk usage, because Essbase stores fewer calculated data values. Database size and index size are also reduced.
- Database restructure time. For example, adding or deleting a Dynamic Calc member in a dense dimension does not change the data block size, so Essbase does not need to restructure the database. See “Restructuring Databases” on page 432.
- Time required to back up the database. Because database size is reduced, Essbase takes less time to perform a backup.

Data values that Essbase calculates dynamically can take longer to retrieve. You can estimate the retrieval time for dynamically calculated members. See “Reducing the Impact on Retrieval Time” on page 428.

Using Dynamic Calculation

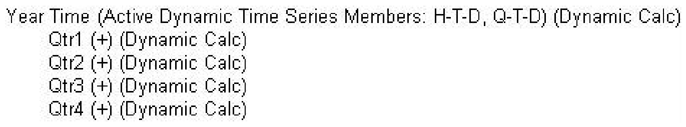
You can tag any member as Dynamic Calc or Dynamic Calc and Store, except the following members:

- Level 0 members that do not have a formula
- Label-only members
- Shared members

Which members you choose to calculate dynamically depends on the database structure and on the balance between (1) the need for reduced calculation time and disk usage and (2) the need for speedy data retrieval for users. See "Choosing Values to Calculate Dynamically" on page 420.

In Outline Editor, you can see which members are Dynamic Calc and which are Dynamic Calc and Store. Figure 128 shows Dynamic Calc members.

Figure 128 Sample.Basic Outline Showing Dynamic Calc Members



```
Year Time (Active Dynamic Time Series Members: H-T-D, Q-T-D) (Dynamic Calc)
    Qtr1 (+) (Dynamic Calc)
    Qtr2 (+) (Dynamic Calc)
    Qtr3 (+) (Dynamic Calc)
    Qtr4 (+) (Dynamic Calc)
```

In Spreadsheet Add-in or Smart View, users can display visual cues to distinguish dynamically calculated values. See the *Oracle Essbase Spreadsheet Add-in Online Help* and the *Oracle Hyperion Smart View for Office Online Help*.

When developing spreadsheets that include dynamically calculated values, spreadsheet designers may want to use the spreadsheet Navigate Without Data option, so that Essbase does not dynamically calculate and store values while test spreadsheets are built.

Choosing Values to Calculate Dynamically

Dynamically calculating some data values decreases calculation time and disk usage and reduces database restructure time but increases retrieval time for dynamically calculated data values.

Use the guidelines described in the following sections when deciding which members to calculate dynamically.

Dense Members and Dynamic Calculation

Consider making the following changes to members of dense dimensions:

- Tag upper-level members of dense dimensions as Dynamic Calc.
- Try tagging level 0 members of dense dimensions with simple formulas as Dynamic Calc, and assess the increase in retrieval time.

Simple formulas do not require Essbase to perform an expensive calculation. Formulas containing financial functions or cross-dimensional operators (->) are complex formulas.

- Do not tag members of dense dimensions as Dynamic Calc and Store.

Sparse Members and Dynamic Calculation

Consider making the following changes to members of sparse dimensions:

- Tag some upper-level members of sparse dimensions that have six or fewer children as Dynamic Calc or Dynamic Calc and Store.
- Tag sparse-dimension members with complex formulas as Dynamic Calc or Dynamic Calc and Store.

  A complex formula requires Essbase to perform an expensive calculation. For example, any formula that contains a financial function is a complex formula. See "Using Complex Formulas" on page 875.
- Tag upper-level members in a dimension that you frequently restructure as Dynamic Calc or Dynamic Calc and Store.
- Do not tag upper-level, sparse-dimension members that have 20 or more descendants as Dynamic Calc or Dynamic Calc and Store.

See "Choosing Between Dynamic Calc and Dynamic Calc and Store" on page 423.

Two-Pass Members and Dynamic Calculation

To reduce the time needed to perform batch calculations, tag two-pass members as Dynamic Calc. You can tag any Dynamic Calc or Dynamic Calc and Store member as two-pass, even if it is not on an accounts dimension. See "Using Two-Pass Calculation" on page 883.

For information about the interaction of members tagged as two-pass and attribute members, see "Comparing Attribute and Standard Dimensions" on page 165.

Parent-Child Relationships and Dynamic Calculation

If a parent member has one child member, and you tag the child as Dynamic Calc, you must also tag the parent as Dynamic Calc. Similarly, if you tag the child as Dynamic Calc and Store, you must also tag the parent as Dynamic Calc and Store. However, if a parent member has one child member, and the parent is a Dynamic Calc or Dynamic Calc and Store member, you do not have to tag the child as Dynamic Calc or Dynamic Calc and Store.

Calculation Scripts and Dynamic Calculation

When Essbase calculates a CALC ALL or CALC DIM statement in a calculation script, it bypasses the calculation of Dynamic Calc and Dynamic Calc and Store members.

Similarly, if a member set function (for example, @CHILDREN or @SIBLINGS) is used to specify the list of members to calculate, Essbase bypasses the calculation of any Dynamic Calc or Dynamic Calc and Store members in the resulting list.

If you specify a Dynamic Calc or Dynamic Calc and Store member explicitly in a calculation script, the calculation script fails. You cannot do a calculation script calculation of a Dynamic Calc or Dynamic Calc and Store member. To use a calculation script to calculate a member explicitly, do not tag the member as Dynamic Calc.

For example, the following calculation script is valid only if Qtr1 is not a Dynamic Calc member:

```
FIX (East, Colas)
   Qtr1;
ENDFIX
```

Formulas and Dynamically Calculated Members

You can include a dynamically calculated member in a formula when you apply the formula to the database outline. For example, if Qtr1 is a Dynamic Calc member, you can place the following formula on Qtr1 in the database outline:

```
Qtr1 = Jan + Feb;
```

You cannot make a dynamically calculated member the target of a formula calculation in a calculation script; Essbase does not reserve memory for a dynamically calculated value and, therefore, cannot assign a value to it. For example, if Qtr1 is a Dynamic Calc or Dynamic Calc and Store member, Essbase displays a syntax error if you include the following formula in a calculation script:

```
Qtr1 = Jan + Feb;
```

If Qtr1 is a Dynamic Calc or Dynamic Calc and Store member and Year is neither Dynamic Calc nor Dynamic Calc and Store, you can use the following formula in a calculation script:

```
Year = Qtr1 + Qtr2;
```

This formula is valid because Essbase does not assign a value to the dynamically calculated member.

Note:

When you reference a dynamically calculated member in a formula in the database outline or in a calculation script, Essbase interrupts the regular calculation to do the dynamic calculation. This interruption can significantly reduce calculation performance.

Scripts and Dynamically Calculated Members

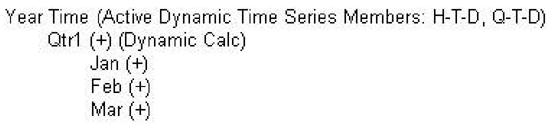
The preprocessing phase of a calculation script cannot determine whether an outline contains dense Dynamic Calc members. If a script contains runtime-dependent formulas, Essbase must calculate all dense Dynamic Calc members when the script is executed. Using the SET FRMLRTDYNAMIC OFF calculation command improves performance by stopping calculation of these Dynamic Calc members. See the *Oracle Essbase Technical Reference*.

Dynamically Calculated Children

If the calculation of a member depends on the calculation of Dynamic Calc or Dynamic Calc and Store child members, Essbase must calculate the child members first during the batch database calculation in order to calculate the parent. Therefore, regular calculation time is not reduced. This requirement applies to members of sparse dimensions and members of dense dimensions.

For example, in Figure 129, Qtr1 is a Dynamic Calc member. Its children, Jan, Feb, and Mar, are not dynamic members. Its parent, Year, is not a dynamic member. When Essbase calculates Year during a batch database calculation, it must consolidate the values of its children, including Qtr1. Therefore, it must take the additional time to calculate Qtr1, although Qtr1 is a Dynamic Calc member.

Figure 129 Sample.Basic Outline, Showing Qtr1 as a Dynamic Calc Member



```
Year Time (Active Dynamic Time Series Members: H-T-D, Q-T-D)
    Qtr1 (+) (Dynamic Calc)
        Jan (+)
        Feb (+)
        Mar (+)
```

Choosing Between Dynamic Calc and Dynamic Calc and Store

In most cases, you can optimize calculation performance and reduce disk usage by using Dynamic Calc members instead of Dynamic Calc and Store members. However, in specific situations, using Dynamic Calc and Store members is optimal.

Recommendations for Sparse Dimension Members

In most cases, to calculate a sparse dimension member dynamically, tag the member as Dynamic Calc instead of Dynamic Calc and Store. When Essbase calculates data values for a member combination that includes a Dynamic Calc member, Essbase calculates only the requested values of the relevant data block. These values can be a subset of the data block.

However, when Essbase calculates data values for a member combination that includes a Dynamic Calc and Store member, Essbase must calculate and store the whole data block, even if the requested data values are a subset of the data block. Thus, the calculation takes longer and the initial retrieval time is greater.

Essbase stores only the data blocks that contain the requested data values. If Essbase must calculate intermediate data blocks to calculate the requested data blocks, it does not store the intermediate blocks.

Calculating the intermediate data blocks can significantly increase the initial retrieval time. For example, in the Sample.Basic database, Market and Product are the sparse dimensions. Assume that Market and the children of Market are Dynamic Calc and Store members. When a user retrieves the data value for the member combination Market -> Cola -> Jan -> Actual -> Sales, Essbase calculates and stores the Market -> Cola data block. To calculate and store Market ->

Cola, Essbase calculates the intermediate data blocks—East -> Cola, West -> Cola, South -> Cola, and Central -> Cola. Essbase does not store these intermediate data blocks.

Recommendations for Members with Specific Characteristics

Using Dynamic Calc and Store may slow initial retrieval; however, subsequent retrievals are faster than for Dynamic Calc members. Use Dynamic Calc and Store instead of Dynamic Calc for the following members:

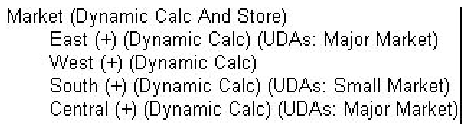
- An upper-level sparse dimension member with children on a remote database.

  Essbase must retrieve the value from the remote database, which increases retrieval time. See "Dynamically Calculating Data in Partitions" on page 433.
- A sparse dimension member with a complex formula.

  A complex formula requires Essbase to perform an expensive calculation. Any formula that contains a financial function or a cross-dimensional member is a complex formula.
- If users frequently retrieve an upper-level member of a sparse dimension, speedy retrieval is important.

For example, in the Sample.Basic database, if most users retrieve data at the Market level, you probably want to tag Market as Dynamic Calc and Store and its children as Dynamic Calc.

Figure 130 Sample.Basic Outline, Market is Dynamic Calc and Store Member



```
Market (Dynamic Calc And Store)
    East (+) (Dynamic Calc) (UDAs: Major Market)
    West (+) (Dynamic Calc)
    South (+) (Dynamic Calc) (UDAs: Small Market)
    Central (+) (Dynamic Calc) (UDAs: Major Market)
```

Recommendations for Dense Dimension Members

Use Dynamic Calc members for dense dimension members. Defining members as Dynamic Calc and Store on a dense dimension provides only a small decrease in retrieval time and in batch calculation time. In addition, database size (disk usage) does not decrease significantly because Essbase reserves space in the data block for the data values of the members.

Recommendations for Data with Many Concurrent Users

Use Dynamic Calc members for data with concurrent users. If many users are concurrently retrieving Essbase data, the initial retrieval time for Dynamic Calc and Store members can be significantly longer than for Dynamic Calc members.

Dynamic Calc and Store member retrieval time increases as the number of concurrent user retrievals increases. However, Dynamic Calc member retrieval time does not increase as concurrent user retrievals increase.

If many users are concurrently accessing data, you may see significantly faster retrieval times if you use Dynamic Calc members instead of Dynamic Calc and Store members.

Understanding How Dynamic Calculation Changes Calculation Order

Using dynamically calculated data values changes the order in which Essbase calculates the values and can have implications for how you administer a database.

Calculation Order for Dynamic Calculation

When Essbase dynamically calculates data values, it calculates the data in an order different from the batch database calculation order.

During batch calculations, Essbase calculates the database in the following order:

1. Dimension tagged as accounts
2. Dimension tagged as time
3. Other dense dimensions (in the order in which they appear in the database outline)
4. Other sparse dimensions (in the order in which they appear in the database outline)
5. Two-pass calculations

See Chapter 24, "Defining Calculation Order."

For dynamically calculated values, on retrieval, Essbase calculates the values by calculating the database in the following order:

1. Sparse dimensions
   - If the dimension tagged as time is sparse and the database outline uses time series data, Essbase bases the sparse calculation on the time dimension.
   - Otherwise, Essbase bases the calculation on the dimension that it normally uses for a batch calculation.
2. Dense dimensions
   a. Dimension tagged as accounts, if dense
   b. Dimension tagged as time, if dense
   c. Time series calculations
   d. Remaining dense dimensions
   e. Two-pass calculations
   f. Attributes

If your data retrieval uses attribute members, the last step in the calculation order is the summation of the attributes. However, the use of attribute members in your query causes Essbase to disregard the value of the Time Balance member in the dynamic calculations. During retrievals that do not use attributes, the value of the Time Balance member is applied to the calculations.

The difference in calculation procedure between the use and nonuse of attribute members generates different results for any upper-level time members that are dynamically calculated.

During retrievals that do not use attributes, these dynamically calculated members are calculated in the last step and, therefore, apply the time balance functionality properly. However, during retrievals that do use attributes, the summation of the attribute is the last step applied. The difference in calculation order produces two different, predictable results for upper-level time members that are dynamically calculated.

Calculation Order for Dynamically Calculating Two-Pass Members

Consider the following information to ensure that Essbase produces the required calculation result when it dynamically calculates data values for members tagged as two-pass (see "Using Two-Pass Calculation" on page 883).

If more than one Dynamic Calc or Dynamic Calc and Store dense dimension member is tagged as two-pass, Essbase performs the dynamic calculation in the first pass, and then calculates the two-pass members in this order:

1. Two-pass members in the accounts dimension, if any exist
2. Two-pass members in the time dimension, if any exist
3. Two-pass members in the remaining dense dimensions in the order in which the dimensions appear in the outline

For example, in the Sample.Basic database, assume the following:

- Margin% in the dense Measures dimension (the dimension tagged as accounts) is tagged as Dynamic Calc and two-pass.
- Variance in the dense Scenario dimension is tagged as Dynamic Calc and two-pass.

Essbase calculates the accounts dimension member first. So, Essbase calculates Margin% (from the Measures dimension) and then calculates Variance (from the Scenario dimension).

If Scenario is a sparse dimension, Essbase calculates Variance first, following the regular calculation order for dynamic calculations. Essbase then calculates Margin%. See "Calculation Order for Dynamic Calculation" on page 425.

This calculation order does not produce the required result, because Essbase needs to calculate Margin % -> Variance using the formula on Margin %, and not the formula on Variance. You can avoid this problem by making Scenario a dense dimension. This problem does not occur if the Measures dimension (the accounts dimension) is sparse, because Essbase still calculates Margin% first.

Calculation Order for Asymmetric Data

Because the calculation order used for dynamic calculations differs from the calculation order used for batch database calculations, in some database outlines, you may get different calculation results if you tag certain members as Dynamic Calc or Dynamic Calc and Store. These differences happen when Essbase dynamically calculates asymmetric data.

Symmetric data calculations produce the same results no matter which dimension is calculated.

Using the data set in Table 48, the calculation for Qtr1-> Profit produces the same result whether you calculate along the dimension tagged as time or the dimension tagged as accounts. Calculating along the time dimension, add the values for Jan, Feb, and Mar:

```
50+100+150=300
```

Calculating along the accounts dimension, subtract Qtr1 -> COGS from Qtr1 -> Sales:

```
600–300=300
```

Table 48 Example of a Symmetric Calculation

Time -> Accounts	Jan	Feb	Mar	Qtr1
Sales	100	200	300	600
COGS	50	100	150	300
Profit (Sales – COGS)	50	100	150	300

Asymmetric data calculations calculate differently along different dimensions.

Using the data set in Table 49, the calculation for East -> Sales produces the correct result when you calculate along the Market dimension, but produces an incorrect result when you calculate along the accounts dimension. Calculating along the Market dimension, adding the values for New York, Florida, and Connecticut produces the correct results:

```
50 + 100 + 100 = 250
```

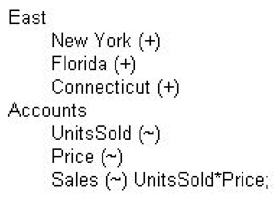
Calculating along the accounts dimension, multiplying the value East -> Price by the value East -> UnitsSold produces incorrect results:

```
15 * 50 = 750
```

Table 49 Example of an Asymmetric Calculation

Market -> Accounts	New York	Florida	Connecticut	East
UnitsSold	10	20	20	50
Price	5	5	5	15
Sales (Price * UnitsSold)	50	100	100	250

In the following outline, East is a sparse dimension, and Accounts is a dense dimension:



```
East
     New York (+)
     Florida (+)
     Connecticut (+)
Accounts
     UnitsSold (~)
     Price (~)
     Sales (~) UnitsSold*Price;
```

If East and Sales are tagged as Dynamic Calc, Essbase calculates a different result than it does if East and Sales are not tagged as Dynamic Calc.

If East and Sales are not Dynamic Calc members, Essbase produces the correct result by calculating these dimensions:

1. Dense Accounts dimension—calculating the values for UnitsSold, Price, and Sales for New York, Florida, and Connecticut
2. Sparse East dimension—aggregating the calculated values for UnitsSold, Price, and Sales for New York, Florida, and Connecticut to obtain the Sales values for East

If East and Sales are Dynamic Calc members, Essbase produces an incorrect result by calculating these dimensions:

1. Sparse East dimension—aggregating the values for UnitsSold, Price, and Sales for New York, Florida, and Connecticut to obtain the values for East
2. Values for East -> Sales—taking the aggregated values in the East data blocks and performing a formula calculation with these values to obtain the value for Sales

To avoid this problem and ensure that you obtain the required results, do not tag the Sales member as Dynamic Calc or Dynamic Calc and Store.

Reducing the Impact on Retrieval Time

The increase in retrieval time when you dynamically calculate a member of a dense dimension is not significant unless the member contains a complex formula. The increase in retrieval time may be significant when you tag members of sparse dimensions as Dynamic Calc or Dynamic Calc and Store.

The following sections discuss ways you can analyze and manage the effect of Dynamic Calc members on a database.

Note:

For a list of functions that have the most significant effect on query retrieval, see "Choosing Between Member Set Functions and Performance" on page 891.

Displaying a Retrieval Factor

To help you estimate any increase in retrieval time, Essbase calculates a retrieval factor for a database outline when you save the outline. Essbase calculates this retrieval factor based on the dynamically calculated data block that is the most expensive for Essbase to calculate. The retrieval factor takes into account only aggregations. It does not consider the retrieval impact of formulas.

The retrieval factor is the number of data blocks that Essbase must retrieve from the disk or from the database to calculate the most expensive block. If the database has Dynamic Calc or Dynamic Calc and Store members in dense dimensions only (no Dynamic Calc or Dynamic Calc and Store members in sparse dimensions), the retrieval factor is 1.

An outline with a high retrieval factor (for example, greater than 2000) can cause long delays when users retrieve data. However, the actual impact on retrieval time also depends on how many dynamically calculated data values a user retrieves. The retrieval factor is only an indicator. In some applications, using Dynamic Calc members may reduce retrieval time because the database size and index size are reduced.

Essbase displays the retrieval factor value in the application log.

➤ To view an estimated retrieval factor, see "Viewing the Essbase Server and Application Logs" on page 742.

A message similar to this sample indicates a retrieval factor:

```
[Wed Sep 20 20:04:13 2000] Local/Sample///Info (1012710)
Essbase needs to retrieve [1] Essbase kernel blocks in order
to calculate the top dynamically-calculated block.
```

This message tells you that Essbase needs to retrieve one block to calculate the most expensive dynamically calculated data block.

Displaying a Summary of Dynamically Calculated Members

When you add Dynamic Calc or Dynamic Calc and Store members to a database outline and save the outline, Essbase provides a summary of how many members are tagged as Dynamic Calc and Dynamic Calc and Store. Essbase displays the summary in the application log.

➤ To view a summary of dynamically calculated members, see "Viewing the Essbase Server and Application Logs" on page 742.

A message similar to this sample is displayed:

```
[Wed Sep 20 20:04:13 2000]Local/Sample///Info(1007125)
The number of Dynamic Calc Non-Store Members = [ 8 6 0 0 2]

[Wed Sep 20 20:04:13 2000]Local/Sample///Info(1007126)
The number of Dynamic Calc Store Members = [ 0 0 0 0 0]
```

This message tells you that there are eight Dynamic Calc members in the first dimension of the database outline, six in the second dimension, and two in the fifth dimension. Dynamic Time Series members are included in this count.

This example does not include Dynamic Calc and Store members.

Increasing Retrieval Buffer Size

When you retrieve data into Spreadsheet Add-in or use Report Writer to retrieve data, Essbase uses the retrieval buffer to optimize the retrieval. Essbase processes the data in sections. Increasing the retrieval buffer size can significantly reduce retrieval time because Essbase can process larger sections of data at one time.

By default, the retrieval buffer size is 10 KB. However, you may speed retrieval time if you set the retrieval buffer size greater than 10 KB. See "Setting the Retrieval Buffer Size" on page 899.

➤ To set the retrieval buffer size, use a tool:

Tool	Topic	Location
Administration Services	Setting the Size of Retrieval Buffers	*Oracle Essbase Administration Services Online Help*
MaxL	**alter database**	*Oracle Essbase Technical Reference*
ESSCMD	SETDBSTATEITEM	*Oracle Essbase Technical Reference*

Using Dynamic Calculator Caches

By default, when Essbase calculates a Dynamic Calc member in a dense dimension (for example, for a query), it writes all blocks needed for the calculation into an area in memory called the dynamic calculator cache. When Essbase writes these blocks into the dynamic calculator cache, it expands them to include all Dynamic Calc members in the dense dimensions.

Using the Essbase dynamic calculator cache enables centralized control of memory usage for dynamic calculations. Managing data blocks in the dynamic calculator cache also reduces the overall memory space requirement and can improve performance by reducing the number of calls to the operating system to do memory allocations.

Note:

The dynamic calculator cache and the calculator cache use different approaches to optimizing calculation performance.

See "Sizing the Calculator Cache" on page 830.

Reviewing Dynamic Calculator Cache Usage

Essbase writes two messages to the application log for each data retrieval. In the following example, the first message describes the total time required for the retrieval:

```
[Thu Aug 03 14:33:00 2005]Local/Sample/Basic/aspen/Info(1001065)
Regular Extractor Elapsed Time : [0.531] seconds

[Thu Aug 03 14:33:00 2005]Local/Sample/Basic/aspen/Info(1001401)
Regular Extractor Big Blocks Allocs -- Dyn.Calc.Cache : [30] non-Dyn.Calc.Cache : [0]
```

If a dynamic calculator cache is used, a second message displays the number of blocks calculated within the data calculator cache (Dyn.Calc.Cache: [*n*]) and the number of blocks calculated in memory outside dynamic calculator cache (non-Dyn.Calc.Cache: [*n*]).

To determine whether the dynamic calculator cache is being used effectively, review both messages and consider your `essbase.cfg` settings. For example, if the message indicates that blocks were calculated outside and in a dynamic calculator cache, you may increase the DYNCALCCACHEMAXSIZE setting. If the specified maximum size is all that you can afford for all dynamic calculator caches on the server, and if using memory outside the calculator cache to complete dynamically calculated retrievals results in unacceptable delays (for example, because of swapping or paging activity), set DYNCALCCACHEWAITFORBLK to TRUE.

You can use the **query database** MaxL statement with the **performance statistics** grammar to view a summary of dynamic calculator cache activity. See the *Oracle Essbase Technical Reference*.

Using Dynamic Calculations with Standard Procedures

Using dynamic calculations with standard Essbase procedures affects these processes:

- Clearing data and data blocks

  You can use the CLEARBLOCK DYNAMIC command to remove data blocks for Dynamic Calc and Store member combinations.

  You can use the CLEARDATA command to mark Dynamic Calc and Store data blocks, so that Essbase knows to recalculate the blocks. The CLEARDATA command has no effect on data values for Dynamic Calc members.

- Copying data

  You cannot copy data to a dynamically calculated data value. You cannot specify a Dynamic Calc or Dynamic Calc and Store member as the target for the DATACOPY calculation command.

- Converting currencies

  You cannot specify a Dynamic Calc or Dynamic Calc and Store member as the target for the CCONV command.

- Loading data

  When you load data, Essbase does not load data into member combinations that contain a Dynamic Calc or Dynamic Calc and Store member. Essbase skips these members during data load and does not display an error message.

  To place data into Dynamic Calc and Dynamic Calc and Store members, after loading data, ensure that Essbase recalculates Dynamic Calc and Store members. See "Effect of Updated Values on Recalculation" on page 418.

- Exporting data

  Essbase does not calculate dynamically calculated values before exporting data. Essbase does not export values for Dynamic Calc members. Essbase exports values for Dynamic Calc and Store members only if a calculated value exists in the database from a previous user retrieval of the data.

- Reporting data

Essbase cannot use the SPARSE data extraction method for dynamically calculated members. The SPARSE data extraction method optimizes performance when a high proportion of the reported data rows are #MISSING. See the <SPARSE command in the *Oracle Essbase Technical Reference*.

- Including dynamic members in calculation scripts

  When calculating a database, Essbase skips the calculation of any Dynamic Calc or Dynamic Calc and Store members. Essbase displays an error message if you attempt to do a member calculation of a Dynamic Calc or Dynamic Calc and Store member in a calculation script. See "Calculation Scripts and Dynamic Calculation" on page 421.

Creating Dynamic Calc and Dynamic Calc and Store Members

➤ To create Dynamic Calc and Dynamic Calc and Store members using Outline Editor, see "Setting Member Storage Properties" in the *Oracle Essbase Administration Services Online Help*.

➤ To create Dynamic Calc and Dynamic Calc and Store members during a dimension build, in the dimension build data file, use the property X for Dynamic Calc and the property V for Dynamic Calc and Store. See "Using the Data Source to Work with Member Properties" on page 275.

Restructuring Databases

When you add a Dynamic Calc member to a dense dimension, Essbase does not reserve space in the data block for the member's values. Therefore, Essbase does not need to restructure the database. However, when you add a Dynamic Calc and Store member to a dense dimension, Essbase does reserve space in the relevant data blocks for the member's values and, therefore, must restructure the database.

When you add a Dynamic Calc or a Dynamic Calc and Store member to a sparse dimension, Essbase updates the index but does not change the relevant data blocks. See "Index Manager" on page 759.

Essbase can save changes to the database outline significantly faster if it does not have to restructure the database.

In the following cases, Essbase does not restructure the database or change the index (Essbase has to save only the database outline, which is very fast):

- Add, delete, or move a dense dimension Dynamic Calc member.

  Essbase does restructure the database if the member is Dynamic Calc and Store.
- Change the storage property of a dense dimension member from Dynamic Calc and Store member to a nondynamic storage property.
- Change the storage property of a sparse dimension Dynamic Calc or Dynamic Calc and Store member to a nondynamic storage property.

- Rename any Dynamic Calc or Dynamic Calc and Store member.

In the following cases, Essbase does not restructure the database but does restructure the database index, which is significantly faster:

- Add, delete, or move sparse dimension Dynamic Calc or Dynamic Calc and Store members.
- Change the storage property of a dense dimension member from a nondynamic value to Dynamic Calc and Store.

In the following cases, Essbase restructures the database:

- Add, delete, or move a dense dimension Dynamic Calc and Store member.

  Essbase does not restructure the database if the member is Dynamic Calc.
- Change a dense dimension Dynamic Calc and Store member to a Dynamic Calc member.
- Change a dense dimension Dynamic Calc member to a Dynamic Calc and Store member.
- Change the storage property of a nondynamic member in a dense dimension to Dynamic Calc.
- Change the storage property of a dense dimension from Dynamic Calc member to a nondynamic value.
- Change the storage property of a nondynamic member in a sparse dimension Dynamic Calc or Dynamic Calc and Store.

See "Types of Database Restructuring" on page 842.

Dynamically Calculating Data in Partitions

You can define Dynamic Calc and Dynamic Calc and Store members in transparent, replicated, or linked regions of the partitions. See Chapter 14, "Designing Partitioned Applications".

For example, if you tag an upper-level, sparse dimension member with children that are on a remote database (transparent database partition) as Dynamic Calc and Store, because Essbase retrieves child values from the other database, retrieval time is increased. You can use Dynamic Calc instead of Dynamic Calc and Store; however, the impact on subsequent retrieval time might be too great.

For example, assume that the local database is the Corporate database, which has transparent partitions to the regional data for East, West, South, and Central. You can tag the parent member Market as Dynamic Calc and Store.

In a transparent partition, the definition on the remote database takes precedence over any definition on the local database. For example, if a member is tagged as Dynamic Calc in the local database but not in the remote database, Essbase retrieves the value from the remote database and does not do the local calculation.

If you are using a replicated partition, consider using Dynamic Calc members instead of Dynamic Calc and Store members. When calculating replicated data, Essbase does not retrieve the child blocks from the remote database; therefore, the impact on retrieval time is not great.

Note:

When Essbase replicates data, it checks the time stamp on each source data block and each corresponding target data block. If the source data block is more recent, Essbase replicates the data in the data block. However, for dynamically calculated data, data blocks and time stamps do not exist. Therefore, Essbase always replicates dynamically calculated data.

27

Calculating Time Series Data

In This Chapter

The information in this chapter applies only to block storage databases and is not relevant to aggregate storage databases. Also see Chapter 57, “Comparison of Aggregate and Block Storage.”

Introduction

Time series calculations assume that you have Dynamic Time Series members defined in the outline. Calculating time series data is helpful in tracking inventory by calculating the first and last values for a time period, and in calculating period-to-date values.

Calculating First, Last, and Average Values

Using time balance and variance reporting tags on the dimension tagged as accounts, you can tell Essbase how to perform time balance calculations on accounts data.

Essbase usually calculates a dimension tagged as time by consolidating or calculating the formulas on the parent’s children. However, you can use accounts tags, such as time balance and variance reporting tags, to consolidate a different kind of value. For example, if you tag a parent member in the accounts dimension with a time balance property of First, Essbase calculates the member by consolidating the value of the member’s first child. For example, in the Sample.Basic database, the Opening Inventory member in the Measures dimension (the accounts dimension) has a time balance property of First. This member represents the inventory at the beginning of the time period. If the time period is Qtr1, Opening Inventory represents the inventory available at the beginning of Jan (the first member in the Qtr1 branch).

To use accounts tags, you must have a dimension tagged as accounts and a dimension tagged as time. You use the First, Last, and Average tags (time balance properties) and the Expense tag (variance reporting property) only on members of a dimension tagged as accounts. The dimensions you tag as time and accounts can be either dense or sparse dimensions.

Note:

For cells of time balance account members, a member in any dimension other than the time dimension that is set with the ^ consolidation property is excluded from the Average calculation; the member is, however, included in First and Last calculations.

Note:

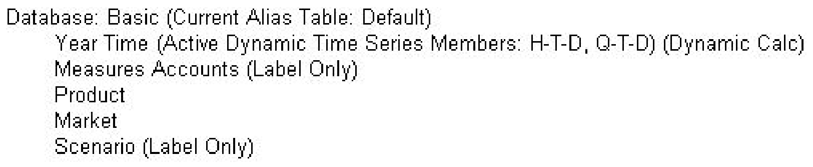
If you are using Intelligent Calculation, changing accounts tags in the database outline does not cause Essbase to restructure the database. You may have to tell Essbase explicitly to recalculate the required data values. See "Changing Formulas and Accounts Properties" on page 415.

Specifying Accounts and Time Dimensions

When you tag a dimension as accounts, Essbase knows that the dimension contains members with accounts tags. When you tag a dimension as time, Essbase knows that this dimension is the one on which to base the time periods for the accounts tags.

As shown in Figure 131, in the Sample.Basic database, the Measures dimension is tagged as accounts, and the Year dimension is tagged as time.

Figure 131 Sample.Basic Outline Showing Accounts and Time Tags



```
Database: Basic (Current Alias Table: Default)
    Year Time (Active Dynamic Time Series Members: H-T-D, Q-T-D) (Dynamic Calc)
    Measures Accounts (Label Only)
    Product
    Market
    Scenario (Label Only)
```

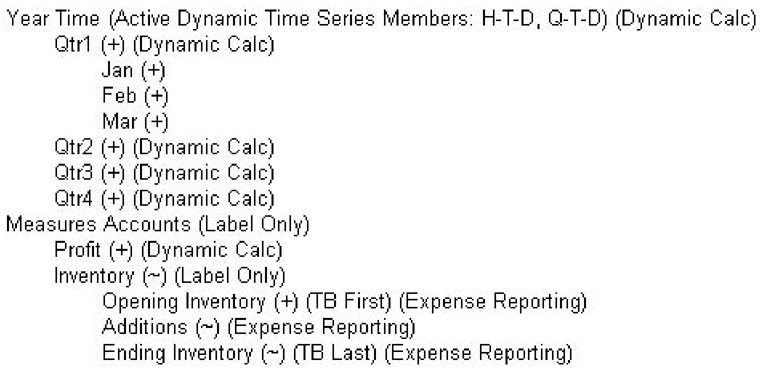
See "Creating a Time Dimension" on page 142 and "Creating an Accounts Dimension" on page 142.

Reporting the Last Value for Each Time Period

For an accounts dimension member, you can tell Essbase to move the last value for each time period up to the next level. To report the last value for each time period, set the member's time balance property as Last. (The tag displays as TB Last in the database outline.)

As shown in Figure 132 on page 437, in the Sample.Basic database, the accounts member Ending Inventory is tagged as TB Last. Ending Inventory consolidates the value for the last month in each quarter and uses that value for that month's parent. For example, the value for Qtr1 is the same as the value for Mar.

Figure 132   Sample.Basic Outline Showing Last Tag



```
Year Time (Active Dynamic Time Series Members: H-T-D, Q-T-D) (Dynamic Calc)
    Qtr1 (+) (Dynamic Calc)
        Jan (+)
        Feb (+)
        Mar (+)
    Qtr2 (+) (Dynamic Calc)
    Qtr3 (+) (Dynamic Calc)
    Qtr4 (+) (Dynamic Calc)
Measures Accounts (Label Only)
    Profit (+) (Dynamic Calc)
    Inventory (~) (Label Only)
        Opening Inventory (+) (TB First) (Expense Reporting)
        Additions (~) (Expense Reporting)
        Ending Inventory (~) (TB Last) (Expense Reporting)
```

For information on tagging an accounts member as Last, see "Setting Time Balance Properties" on page 142.

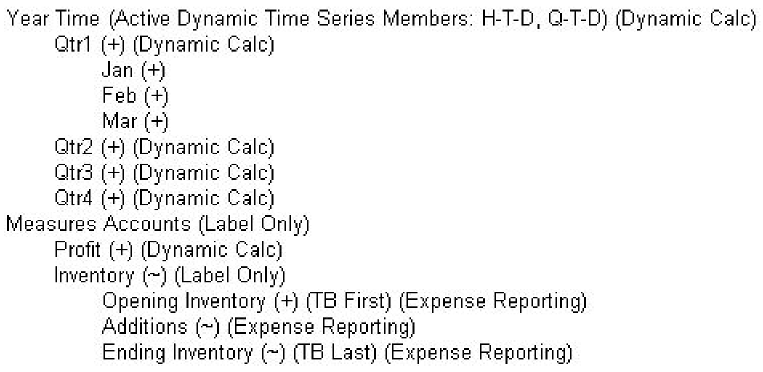
By default, Essbase does not skip #MISSING or zero (0) values when calculating a parent value. You can choose to skip these values. For a discussion of how and why to skip #MISSING values, see "Skipping #MISSING and Zero Values" on page 438.

Reporting the First Value for Each Time Period

For an accounts dimension member, you can tell Essbase to move the first value for each time period up to the next level. To report the first value for each time period, set the member's time balance property as First. (The tag displays as TB First in the database outline.)

As shown in Figure 133, in the Sample.Basic database, the accounts member Opening Inventory is tagged as TB First. Opening Inventory consolidates the value of the first month in each quarter and uses that value for that month's parent. For example, the value for Qtr1 is the same as the value for Jan.

Figure 133   Sample.Basic Outline Showing First Tag



```
Year Time (Active Dynamic Time Series Members: H-T-D, Q-T-D) (Dynamic Calc)
    Qtr1 (+) (Dynamic Calc)
        Jan (+)
        Feb (+)
        Mar (+)
    Qtr2 (+) (Dynamic Calc)
    Qtr3 (+) (Dynamic Calc)
    Qtr4 (+) (Dynamic Calc)
Measures Accounts (Label Only)
    Profit (+) (Dynamic Calc)
    Inventory (~) (Label Only)
        Opening Inventory (+) (TB First) (Expense Reporting)
        Additions (~) (Expense Reporting)
        Ending Inventory (~) (TB Last) (Expense Reporting)
```

For information on tagging an accounts member as First, see "Setting Time Balance Properties" on page 142.

By default, Essbase does not skip #MISSING or zero (0) values when calculating a parent value. You can choose to skip these values. See "Skipping #MISSING and Zero Values" on page 438.

Reporting the Average Value for Each Time Period

For an accounts dimension member, you can tell Essbase to average values across time periods and consolidate the average up to the next level. For example, you can tell Essbase to average the values for Jan, Feb, and Mar and then use that value for the Qtr1 value. To report the average value for each time period, set the member's time balance property as Average.

For information on tagging an accounts member as Average, see "Setting Time Balance Properties" on page 142.

By default, Essbase does not skip #MISSING or zero (0) values when it calculates a parent value. Thus, when it calculates the average, Essbase aggregates the child values and divides by the number of children, regardless of whether the children have #MISSING or zero values. You can tell Essbase to skip #MISSING and zero values. See "Skipping #MISSING and Zero Values" on page 438.

Skipping #MISSING and Zero Values

You can tell Essbase how to treat #MISSING and zero (0) values when doing time balance calculations. A #MISSING value is a marker in Essbase that indicates that the data in this location does not exist, does not contain any meaningful value, or was never entered.

By default, Essbase does not skip #MISSING or 0 (zero) values when calculating a parent value.

You can override this default by setting a skip property. See "Setting Skip Properties" on page 144.

For example, if you tag an accounts dimension member as Last and Skip Missing, then Essbase consolidates the last nonmissing child to the parent. Consider the example in Table 50:

Table 50 Example of the Effects of the Skip Missing

Accounts -> Time	Jan	Feb	Mar	Qtr1
Accounts Member (**Last, Skip Missing**)	60	70	#MI	70

Tagging an account as Average and Skip Missing may produce different results from tagging that account as Average and Skip None. A calculation performed with Average and Skip None produces correct results because no data is skipped. But because grandparents with children are consolidated by summing the averages, results of a calculation on an account with Average and Skip Missing is incorrect unless you use Dynamic Calc or Two-Pass tags.

Considering the Effects of First, Last, and Average Tags

Table 51 shows how Essbase consolidates the time dimension based on the time balance (TB) First, Last, and Average tags on accounts dimension members.

Table 51 Example of the Effects of (TB) First, Last and Average

Accounts -> Time	Jan	Feb	Mar	Qtr1	
Accounts Member1	11	12	13	36	Value of Jan + Feb + Mar
Accounts Member2 (**TB First**)	20	25	21	20	Value of Jan
Accounts Member3 (**TB Last**)	25	21	30	30	Value of Mar
Accounts Member4 (**TB Average**)	20	30	28	26	Average of Jan, Feb, Mar

Placing Formulas on Time and Accounts Dimensions

If you place a member formula on a time or accounts dimension, it may be overwritten by a time balance calculation.

Table 52 shows an example in which Opening Inventory is tagged as First:

Table 52 Example of the Effects of (TB) First

Measures -> Year	Jan	Feb	Mar	Qtr1
Opening Inventory: **First**	30000	28000	27000	30000

Because Opening Inventory is tagged as First, Essbase calculates Opening Inventory for Qtr1 by taking the Opening Inventory for Jan value. Any member formula that is placed on Qtr1 in the database outline is overwritten by this time balance calculation.

Calculating Period-to-Date Values

You can calculate period-to-date values for data. For example, you can calculate the sales values for the current quarter up to the current month. If the current month is May, using a standard calendar quarter, the quarter total is the total of the values for April and May.

In Essbase, you can calculate period-to-date values in two ways:

- During a batch calculation, using the @PTD function
- Dynamically, when a user requests the values, using Dynamic Time Series members

This section explains how to use Dynamic Time Series members to dynamically calculate period-to-date values. Using Dynamic Time Series members is almost always the most efficient method. For an example, see “Calculating Period-to-Date Values” on page 379.

Using Dynamic Time Series Members

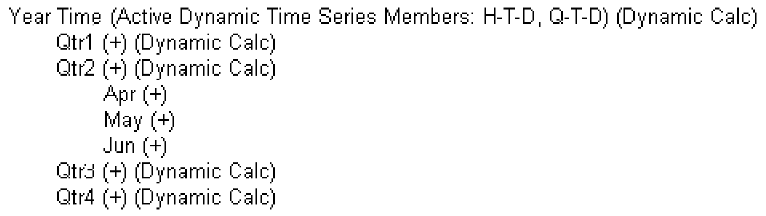
To calculate period-to-date values dynamically, you must use a Dynamic Time Series member for a period on the dimension tagged as time. See “Specifying Accounts and Time Dimensions” on page 436.

You do not create the Dynamic Time Series member directly in the database outline. Instead, you enable a predefined Dynamic Time Series member and associate it with an appropriate generation number.

For example, to calculate quarter-to-date values, you enable the Q-T-D member and associate it with the generation to which you want to apply the Dynamic Time Series member. In Sample.Basic, the generation containing quarters is generation number 2, which contains the Qtr1, Qtr2, Qtr3, and Qtr4 members. Essbase creates a Dynamic Time Series member called Q-T-D and associates it with generation 2. The Q-T-D member calculates monthly values up to the current month in the quarter. see "Enabling Dynamic Time Series Members" on page 441.

Dynamic Time Series members are not displayed as members in the database outline. Instead, Essbase lists the currently active Dynamic Time Series members in a comment on the time dimension. In the outline in Figure 134, H-T-D (history-to-date) and Q-T-D (quarter-to-date) are active. H-T-D is associated with generation 1; Q-T-D is associated with generation 2.

Figure 134 Sample.Basic Outline Showing Dynamic Time Series



```
Year Time (Active Dynamic Time Series Members: H-T-D, Q-T-D) (Dynamic Calc)
    Qtr1 (+) (Dynamic Calc)
    Qtr2 (+) (Dynamic Calc)
        Apr (+)
        May (+)
        Jun (+)
    Qtr3 (+) (Dynamic Calc)
    Qtr4 (+) (Dynamic Calc)
```

Essbase provides eight predefined Dynamic Time Series members:

- HTD (history-to-date)
- Y-T-D (year-to-date)
- S-T-D (season-to-date)
- P-T-D (period-to-date)
- Q-T-D (quarter-to-date)
- M-T-D (month-to-date)
- W-T-D (week-to-date)
- D-T-D (day-to-date)

These members provide up to eight levels of period-to-date reporting. How many and which members you use depends on the data and the database outline.

For example, if the database contains hourly, daily, weekly, monthly, quarterly, and yearly data, you can report day-to date (D-T-D), week-to-date (W-T-D), month-to-date (M-T-D), quarter-to-date (Q-T-D), and year-to-date (Y-T-D) information.

If the database contains monthly data for the last five years, you can report year-to-date (Y-T-D) and history-to-date (H-T-D) information, up to a specific year.

If the database tracks data for seasonal time periods, you can report period-to-date (P-T-D) or season-to-date (S-T-D) information.

You can associate a Dynamic Time Series member with any generation in the time dimension except the highest generation number, regardless of the data. For example, you can use the P-T-D member to report quarter-to-date information. You cannot associate Dynamic Time Series members with level 0 members of the time dimension.

Note:

Oracle recommends that you avoid assigning time balance properties (First, Last, Average, Skip Missing) to members set for dynamic calculations if you plan to use these members in Dynamic Time Series calculations. Doing so may retrieve incorrect values for the parent members in your accounts dimension.

Enabling Dynamic Time Series Members

To use Dynamic Time Series members, you must enable them. If required, you can specify aliases for Dynamic Time Series members. See "Specifying Alias Names for Dynamic Time Series Members" on page 441.

➤ To enable Dynamic Time Series members, see "Enabling Dynamic Time Series Members" in the *Oracle Essbase Administration Services Online Help*.

Note:

The number of generations displayed depends on the number of generations in the time dimension. You cannot associate Dynamic Time Series members with the highest generation (level 0 members).

After you enable Dynamic Time Series members in the database outline, Essbase adds a comment to the dimension tagged as time; for example, the Year dimension from Sample.Basic showing H-T-D and Q-T-D defined:

```
Year Time (Active Dynamic Time Series Members: H-T-D, Q-T-D) (Dynamic Calc)
```

Disabling Dynamic Time Series Members

To disable a Dynamic Time Series member, tell Essbase not to use the predefined member.

➤ To disable Dynamic Time Series members, see "Disabling Dynamic Time Series Members" in the *Oracle Essbase Administration Services Online Help*.

Specifying Alias Names for Dynamic Time Series Members

You can specify alias names for predefined Dynamic Time Series members, such as QtrToDate, for the Q-T-D Dynamic Time Series member. You can then use the alias names to retrieve the Dynamic Time Series members in Smart View, Spreadsheet Add-in, or in a report.

You can create up to eight alias names for each Dynamic Time Series member. Essbase saves each alias name in the Dynamic Time Series alias table that you specify.

➤ To create aliases for Dynamic Time Series members, see "Creating Aliases for Dynamic Time Series Members" in the *Oracle Essbase Administration Services Online Help*.

For information on specifying and displaying alias names, see "Setting Aliases" on page 152.

Applying Predefined Generation Names to Dynamic Time Series Members

When you enable a Dynamic Time Series member and associate it with a generation number, Essbase creates a predefined generation name for that generation number. See "Naming Generations and Levels" on page 157.

➤ To display generation and level names, see "Naming Generations and Levels" in the *Oracle Essbase Administration Services Online Help*.

Table 53 lists the Dynamic Time Series members and their corresponding generation names:

Table 53 Dynamic Time Series Members and Corresponding Generation Names

Member	Generation Name
D-T-D	Day
H-T-D	History
M-T-D	Month
P-T-D	Period
Q-T-D	Quarter
S-T-D	Season
W-T-D	Week
Y-T-D	Year

These member and generation names are reserved for use by Essbase. If you use one of these generation names to create a generation name on the time dimension, Essbase automatically creates and enables the corresponding Dynamic Time Series member for you.

For example, in Sample.Basic, you can create a generation name called Quarter for generation number 2. Quarter contains quarterly data in the members Qtr1, Qtr2, and so on. When you create the generation name Quarter, Essbase creates and enables a Dynamic Time Series member called Q-T-D.

Retrieving Period-to-Date Values

When you retrieve a Dynamic Time Series member, you must tell Essbase the time period up to which you want to calculate the period-to-date value. This time period, known as the *latest time period,* must be a level 0 member on the time dimension.

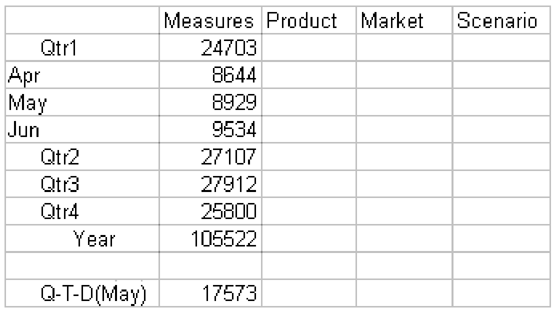
➤ Use the following methods to specify the latest time period:

- For a specific member, in Smart View or Spreadsheet Add-in, specify the latest period member name. Place that name after the Dynamic Time Series member or alias name. For example, Q-T-D(May) returns the quarter-to-date value by adding values for April and May.
- For a retrieval, use one of the following methods to specify the latest time period:
  - Use the <LATEST command in Report Writer.
  - Specify the Latest Time Period option in the Essbase Options dialog box in Spreadsheet Add-in.

    The member-specific setting—for example, Q-T-D(May)—takes precedence over the <LATEST or Latest Time Series option setting.

In the example in Figure 135, Q-T-D(May) displays the period-to-date value for May that is obtained by adding the values for Apr and May (8644 + 8929 = 17573).

Figure 135 Spreadsheet Showing Period-To-Date Value for May



	Measures	Product	Market	Scenario
Qtr1	24703			
Apr	8644			
May	8929			
Jun	9534			
Qtr2	27107			
Qtr3	27912			
Qtr4	25800			
Year	105522			
Q-T-D(May)	17573			

Using Dynamic Time Series Members in Transparent Partitions

To optimize query time across transparent partitions for outlines containing Dynamic Time Series members, use the `essbase.cfg` setting TARGETTIMESERIESOPT.

See TARGETTIMESERIESOPT in the *Oracle Essbase Technical Reference* and Chapter 15, "Creating and Maintaining Partitions,"

28

Developing Calculation Scripts

In This Chapter

The information in this chapter applies only to block storage databases and is not relevant to aggregate storage databases.

Also see:

- Chapter 29, "Reviewing Examples of Calculation Scripts"
- Chapter 22, "Developing Formulas"
- "Using Essbase to Manage Artifacts" on page 718
- Chapter 57, "Comparison of Aggregate and Block Storage."

Understanding Calculation Scripts

A calculation script, which contains a series of calculation commands, equations, and formulas, allows you to define calculations other than those defined by the database outline.

In a calculation script, you can perform a default calculation (CALC ALL) or a calculation of your choosing (for example, you can calculate part of a database or copy data values between members). You must write a calculation script to do any of the following tasks:

- Calculate a subset of a database

  See "Calculating a Subset of a Database" on page 459.
- Change the calculation order of the dense and sparse dimensions in a database
- Perform a complex calculation in a specific order or perform a calculation that requires multiple iterations through the data (for example, some two-pass calculations require a calculation script)
- Perform any two-pass calculation on a dimension without an accounts tag

  See "Using Two-Pass Calculation" on page 883.
- Perform a currency conversion

See Chapter 13, “Designing and Building Currency Conversion Applications.”

- Calculate member formulas that differ from formulas in the database outline (formulas in a calculation script override formulas in the database outline)
- Use an API interface to create a custom calculation dynamically
- Use control of flow logic in a calculation (for example, to use the IF…ELSE…ENDIF or the LOOP…ENDLOOP commands)
- Clear or copy data from specific members

  See “Copying Data” on page 459.
- Define temporary variables for use in a database calculation

  See “Declaring Data Variables” on page 450.
- Force a recalculation of data blocks after you have changed a formula or an accounts property on the database outline
- Control how Essbase uses Intelligent Calculation when calculating a database

  See Chapter 25, “Understanding Intelligent Calculation.”

The following calculation script, based on the Sample.Basic database, calculates the Actual values from the Year, Measures, Market, and Product dimensions:

```
FIX (Actual)
   CALC DIM(Year, Measures, Market, Product);
ENDFIX
```

Using Calculation Script Editor in Administration Services Console, you can create calculation scripts by:

- Entering the calculation script in the text area of the script editor
- Using the UI features of the script editor to build the script
- Creating the script in the text editor and pasting the script text into Calculation Script Editor

See “About Calculation Script Editor” in the *Oracle Essbase Administration Services Online Help*.

Calculation scripts created using Administration Services are given a `.csc` extension by default. If you run a calculation script from Administration Services, Smart View, or Spreadsheet Add-in, the file must have a `.csc` extension. However, because a calculation script is a text file, you can use MaxL or ESSCMD to run any text file as a calculation script.

A calculation script can also be a string defined in memory and accessed through the API on an Essbase client or an Essbase Server. Therefore, from dialog boxes, you can dynamically create a calculation script that is based on user selections.

Understanding Calculation Script Syntax

Essbase provides a flexible set of commands that you can use to control how a database is calculated. You can construct calculation scripts from commands and formulas. In Calculation Script Editor, script elements are color-coded to aid readability and you can enable autocompletion to help build scripts interactively as you type. See “About Calculation Script Editor” in the *Oracle Essbase Administration Services Online Help*.

Computation, control of flow, and data declarations are discussed in the following sections.

For a full list of calculation script commands and syntax, see the *Oracle Essbase Technical Reference*.

Understanding the Rules for Calculation Script Syntax

When you create a calculation script, you must apply the following rules:

- End each formula or calculation script command with a semicolon (;):

  Example 1:

  ```
  CALC DIM(Product, Measures);
  ```

  Example 2:

  ```
  DATACOPY Plan TO Revised_Plan;
  ```

  Example 3:

  ```
  "Market Share" = Sales % Sales -> Market;
  ```

  Example 4:

  ```
  IF (Sales <> #MISSING)
     Commission = Sales * .9;
     ELSE
        Commission = #MISSING;
  ENDIF;
  ```

  You do not need to end the following commands with semicolons:

  ```
  IF
  ENDIF
  ELSE
  ELSIF
  FIX
  ENDFIX
  EXCLUDE
  ENDEXCLUDE
  LOOP
  ENDLOOP
  ```

  Note:

  Although not required, it is good practice to follow each ENDIF statement in a formula with a semicolon.

- Enclose a member name in double quotation marks (" "), if that member name meets any of the following conditions:
  - Contains spaces. For example:

```
"Opening Inventory" = "Ending Inventory" - Sales + Additions;
```

  - Is the same as an operator, function name, or keyword.

    See "Using Dimension and Member Names in Calculation Scripts, Report Scripts, Formulas, Filters, Substitution Variable Values and Environment Variable Values" on page 1063.

  - Includes any nonalphanumeric character; for example, hyphen (-), asterisk (*), or slash (/).

    See "Using Dimension and Member Names in Calculation Scripts, Report Scripts, Formulas, Filters, Substitution Variable Values and Environment Variable Values" on page 1063.

  - Contains only numerals or starts with a numeral; for example, "100" or "10Prod".
  - Begins with an ampersand (&). The leading ampersand (&) is reserved for substitution variables. If a member name begins with &, enclose it in quotation marks. Do not enclose substitution variables in quotation marks in a calculation script.
  - Contains a dot (.); for example, 1999.Jan or .100.

- If you are using an IF statement or an interdependent formula, enclose the formula in parentheses to associate it with the specified member. For example, the following formula is associated with the Commission member in the database outline:

```
Commission
(IF(Sales < 100)
   Commission = 0;
ENDIF;)
```

- End each IF statement in a formula with an ENDIF statement. For example, the previous formula contains a simple IF...ENDIF statement.
- If you are using an IF statement that is nested within another IF statement, end each IF with an ENDIF statement. For example:

```
"Opening Inventory"
(IF (@ISMBR(Budget))
   IF (@ISMBR(Jan))
      "Opening Inventory" = Jan;
   ELSE
      "Opening Inventory" = @PRIOR("Ending Inventory");
   ENDIF;
ENDIF;)
```

- You do not need to end ELSE or ELSEIF statements with ENDIF statements. For example:

```
Marketing
(IF (@ISMBR(@DESCENDANTS(West)) OR @ISMBR(@DESCENDANTS(East)))
   Marketing = Marketing * 1.5;
ELSEIF(@ISMBR(@DESCENDANTS(South)))
   Marketing = Marketing * .9;
ELSE
   Marketing = Marketing * 1.1;
ENDIF;)
```

Note:

If you use ELSE IF (with a space) rather than ELSEIF (one word) in a formula, you must supply an ENDIF for the IF statement.

- End each FIX statement with an ENDFIX statement. For example:

```
FIX(Budget,@DESCENDANTS(East))
   CALC DIM(Year, Measures, Product);
ENDFIX
```

- End each EXCLUDE statement with an ENDEXCLUDE statement.

When you write a calculation script, use the Calculation Script Editor syntax checker to check the syntax. See "Checking Syntax" on page 468.

Understanding Calculation Commands

You can use the calculation commands in Table 54 to perform a database calculation that is based on the structure and formulas in the database outline.

Table 54 List of Commands for Calculating a Database

Calculation	Command
The entire database, based on the outline	CALC ALL
A specified dimension or dimensions	CALC DIM
All members tagged as two-pass on the dimension tagged as accounts	CALC TWOPASS
The formula applied to a member in the database outline, where *membername* is the name of the member to which the formula is applied	*membername*
All members tagged as Average on the dimension tagged as accounts	CALC AVERAGE
All members tagged as First on the dimension tagged as accounts	CALC FIRST
All members tagged as Last on the dimension tagged as accounts	CALC LAST
Currency conversions	CCONV

Controlling the Flow of Calculations

You can use the commands in Table 55 to manipulate the flow of calculations:

Table 55 List of Commands to Control the Flow of Calculations

Calculation	Command
Calculate a subset of a database by inclusion	FIX...ENDFIX
Calculate a subset of a database by exclusion	EXCLUDE...ENDEXCLUDE
Specify the number of times that commands are iterated	LOOP...ENDLOOP

You can also use the IF and ENDIF commands to specify conditional calculations.

Note:

You cannot branch from one calculation script to another calculation script.

Declaring Data Variables

You can use the commands in Table 56 to declare temporary variables and, if required, to set their initial values. Temporary variables store the results of intermediate calculations.

You can also use substitution variables in a calculation script (see "Using Substitution Variables in Calculation Scripts" on page 455).

Table 56 List of Commands for Declaring Data Variables

Calculation	Command
Declare one-dimensional array variables	ARRAY
Declare a temporary variable that contains a single value	VAR

Values stored in temporary variables exist only while the calculation script is running. You cannot report on the values of temporary variables.

Variable and array names are character strings that contain any of the following characters:

- Letters a–z
- Numerals 0–9
- Special characters: $ (dollar sign), # (pound sign), and _ (underscore)

Typically, arrays are used to store variables as part of a member formula. The size of the array variable is determined by the number of members in the corresponding dimension. For example, if the Scenario dimension has four members, the following command creates an array called `Discount` with four entries:

```
ARRAY Discount[Scenario];
```

You can use multiple arrays at a time.

Specifying Global Settings for a Database Calculation

You can use the commands in Table 57 to define calculation behavior:

Table 57 List of Commands for Defining Calculation Behavior

Calculation	Command
Specify how Essbase treats #MISSING values during a calculation	SET AGGMISSG
Adjust the default calculator cache size	SET CACHE

Calculation	Command
Enable parallel calculation (see “Using Parallel Calculation” on page 866)	SET CALCPARALLEL
Increase the number of dimensions used to identify tasks for parallel calculation (see “Using Parallel Calculation” on page 866)	SET CALCTASKDIMS
Optimize the calculation of sparse dimension formulas in large database outlines (see “Optimizing Formulas on Sparse Dimensions in Large Database Outlines” on page 876)	SET FRMLBOTTOMUP

Display messages to trace a calculation	SET MSG
	SET NOTICE

Turn on and turn off Intelligent Calculation (see “Turning Intelligent Calculation On and Off” on page 406)	SET UPDATECALC
Control how Essbase marks data blocks for Intelligent Calculation (see “Using the SET CLEARUPDATESTATUS Command” on page 407)	SET CLEARUPDATESTATUS
Specify the maximum number of blocks that Essbase can lock concurrently when calculating a sparse member formula	SET LOCKBLOCK
Turn on and turn off the Create Blocks on Equation setting (controls creation of blocks when you assign nonconstant values to members of a sparse dimension) (see “Nonconstant Values Assigned to Members in a Sparse Dimension” on page 877)	SET CREATEBLOCKEQ
Enable calculations on potential data blocks and save these blocks when the result is not #MISSING	SET CREATENONMISSINGBLK
(Currency conversions) Restrict consolidations to parents that have the same defined currency (see “Calculating Databases” on page 207)	SET UPTOLOCAL

A SET command in a calculation script stays in effect until the next occurrence of the same SET command.

In the following calculation script, Essbase displays messages at the detail level when calculating the Year dimension and displays messages at the summary level when calculating the Measures dimension:

```
SET MSG DETAIL;
CALC DIM(Year);
SET MSG SUMMARY;
CALC DIM(Measures);
```

Some SET calculation commands trigger additional passes through the database.

In the following calculation script, Essbase calculates member combinations for Qtr1 with SET AGGMISSG turned on, and then does a second calculation pass through the database and calculates member combinations for East with SET AGGMISSG turned off:

```
SET AGGMISSG ON;
Qtr1;
SET AGGMISSG OFF;
East;
```

See the SET AGGMISSG command in the *Oracle Essbase Technical Reference*. Also see "Using Two-Pass Calculation" on page 883.

Adding Comments

You can include comments to annotate calculation scripts. Essbase ignores these comments when it runs the calculation script.

To include a comment, start the comment with /* and end the comment with */. For example:

```
/* This calculation script comment
   spans two lines.  */
```

Planning Calculation Script Strategy

You can enter a calculation script directly into the text area of Calculation Script Editor, or you can use the UI features of Calculation Script Editor to build the calculation script.

Using Formulas in a Calculation Script

You can place member formulas in a calculation script. When you do, the formula overrides conflicting formulas that are applied to members in the database outline.

In a calculation script, you can perform both of these operations:

- Calculate a member formula on the database outline
- Define a formula

To calculate a formula that is applied to a member in the database outline, use the member name followed by a semicolon (;).

For example, the following command calculates the formula applied to the Variance member in the database outline:

```
Variance;
```

To override values that result from calculating an outline, manually apply a formula that you define in a calculation script.

For example, the following formula cycles through the database, adding the values in the members Payroll, Marketing, and Misc, and placing the result in the Expenses member.

```
Expenses = Payroll + Marketing + Misc;
```

This formula overrides any formula placed on the Expenses member in the database outline.

Note:

You cannot apply formulas to shared members or label only members.

See Chapter 22, "Developing Formulas."

Basic Equations

You can use equations in a calculation script to assign value to a member. The syntax for an equation:

```
Member = mathematical expression;
```

Member is a member name from the database outline, and *mathematical expression* is any valid mathematical expression.

Essbase evaluates the expression and assigns the value to the specified member.

For example, the following formula causes Essbase to cycle through the database, subtracting the values in COGS from the values in Sales and placing the result in Margin:

```
Margin = Sales - COGS;
```

The next formula cycles through the database subtracting the values in Cost from the values in Retail, calculating the resulting values as a percentage of the values in Retail, and placing the results in Markup:

```
Markup = (Retail - Cost) % Retail;
```

You can also use the > (greater than) and < (less than) logical operators in equations.

In the following example, if it is true that February sales are greater than January sales, Sales Increase Flag results in a 1 value; if false, the result is a 0 value:

```
Sales Increase Flag = Sales -> Feb > Sales ->
Jan;
```

Conditional Equations

When you use an IF statement as part of a member formula in a calculation script, you must:

- Associate the IF statement with a single member
- Enclose the IF statement in parentheses

In the following example, the entire IF…ENDIF statement is enclosed in parentheses and associated with the Profit member, `Profit (IF(...)...)`:

```
Profit
(IF (Sales > 100)
   Profit = (Sales - COGS) * 2;
ELSE
   Profit = (Sales - COGS) * 1.5;
ENDIF;)
```

Essbase cycles through the database and performs the following calculations:

1. The IF statement checks whether the value of Sales for the current member combination is greater than 100.
2. If Sales is greater than 100, Essbase subtracts the value in COGS from the value in Sales, multiplies the difference by 2, and places the result in Profit.

3. If Sales is less than or equal to 100, Essbase subtracts the value in COGS from the value in Sales, multiplies the difference by 1.5, and places the result in Profit.

Interdependent Formulas

When you use an interdependent formula in a calculation script, the same rules apply as for the IF statement. You must:

- Associate the formula with a single member
- Enclose the formula in parentheses

In the following example, the entire formula is enclosed in parentheses and associated with the Opening Inventory member:

```
"Opening Inventory"
(IF(NOT @ISMBR (Jan))
   "Opening Inventory" = @PRIOR("Ending Inventory");
ENDIF;)
"Ending Inventory" = "Opening Inventory" - Sales + Additions;
```

Using a Calculation Script to Control Intelligent Calculation

Assume that you have a formula on a sparse dimension member, and the formula contains either of the following type of function:

- Relationship (for example, @PRIOR or @NEXT)
- Financial (for example, @NPV or @INTEREST)

Essbase always recalculates the data block that contains the formula, even if the data block is marked as clean for the purposes of Intelligent Calculation.

See "Calculating Data Blocks" on page 409 and Chapter 25, "Understanding Intelligent Calculation."

Grouping Formulas and Calculations

You may achieve significant calculation performance improvements by carefully grouping formulas and dimensions in a calculation script. See the next two sections.

Calculating a Series of Member Formulas

When you calculate formulas, avoid using parentheses unnecessarily.

In the following example, inappropriately placed parentheses cause Essbase to perform two calculation passes through the database: once calculating the formulas on the members Qtr1 and Qtr2; and once calculating the formula on Qtr3:

```
(Qtr1;
Qtr2;)
Qtr3;
```

In contrast, the following configurations cause Essbase to cycle through the database only once, calculating the formulas on the members Qtr1, Qtr2, and Qtr3:

```
Qtr1;
Qtr2;
Qtr3;
```

or

```
(Qtr1;
Qtr2;
Qtr3;)
```

Similarly, the following formulas cause Essbase to cycle through the database once, calculating both formulas in one pass:

```
Profit = (Sales - COGS) * 1.5;
Market = East + West;
```

Calculating a Series of Dimensions

When calculating a series of dimensions, you can optimize performance by grouping the dimensions wherever possible.

For example, the following formula causes Essbase to cycle through the database only once:

```
CALC DIM(Year, Measures);
```

In contrast, the following syntax causes Essbase to cycle through the database twice, once for each CALC DIM command:

```
CALC DIM(Year);
CALC DIM(Measures);
```

Using Substitution Variables in Calculation Scripts

When you include a substitution variable in a calculation script, Essbase replaces the substitution variable with the value you specified for the substitution variable. Substitution variables are useful, for example, when you reference information or lists of members that change frequently.

You create and specify values for substitution values in Administration Services. See "Using Substitution Variables" on page 120.

You can create substitution variables at the server, application, and database levels. To use a substitution variable in a calculation script, the substitution variable must be available to the calculation script. For example, a database-level substitution variable is available only to calculation scripts within the database; a server-level substitution variable is available to any calculation script on the server.

In a calculation script, insert an ampersand (&) before a substitution variable. Essbase treats any string that begins with a leading ampersand as a substitution variable, replacing the variable with its assigned value before parsing the calculation script.

For example, in Sample.Basic, to calculate Qtr1 as the current quarter:

- Create a substitution variable for the current quarter (`&CurQtr`) and assign it the value `Qtr1`
- Create a calculation script that uses the &CurQtr substitution variable

```
FIX(&CurQtr)
   CALC DIM(Measures, Product);
ENDFIX
```

Using Environment Variables in Calculation Scripts

In calculation scripts, you can use system environment variables as placeholders for user-specific system settings. Because environment variables are defined at the operating system level, they are available to all calculation scripts on Essbase Server.

Note:

Environment variables cannot be used in MDX queries.

To declare a system environment variable, see your operating system documentation.

To use an environment variable in a calculation script, insert the dollar sign ($) character before the environment variable name. Essbase treats any string that begins with a leading dollar sign as an environment variable, replacing the variable with its assigned value before parsing the calculation script. If a member name begins with $, enclose the name in quotation marks.

When using environment variables in calculation scripts, follow these guidelines:

- Environment variable names:
  - Must consist of alphanumeric characters or underscores (_)
  - Cannot include nonalphanumeric characters, such as hyphens (-), asterisks (*), and slashes (/)
  - Cannot exceed 320 bytes (for Unicode-mode and non-Unicode mode applications)
- Environment variable values:
  - May contain any character except a leading dollar sign ($)
  - Whether numeric or non-numeric, must be enclosed in quotation marks (" ")—for example:

    ```
    MY_PROD="100"
    ```

    ```
    ENV_FILE="E:\temp\export1.txt"
    ```

    For non-numeric values, if you do not enclose the value in quotation marks when you define the environment variable, Essbase automatically encloses the value with quotation marks when the environment variable is passed.

    For numeric values, Essbase does not automatically enclose the value with quotation marks when the variable is passed. (The reason is that Essbase cannot determine if you intend to pass a numeric value or a member name. For example, if you use a calculation script statement such as 'Sales = $MY_SALES' where MY_SALES=700, the intent is to pass the numeric value of 700. If, however, Essbase encloses MY_SALES in quotation marks, MY_SALES is treated as a member name. The member name would be passed, not the numeric value, causing an error.) If you want the numeric value of the variable to be treated as a string, you must enclose the value with quotation marks when you define the environment variable.

- Cannot exceed 256 bytes (for Unicode-mode and non-Unicode mode applications)

For example, you can use an environment variable to define the path and filename for an export file when exporting a block of data to a flat file. In the following calculation script, the path and filename are explicitly defined (see the text in **bold**):

```
SET DATAEXPORTOPTIONS
{
   DATAEXPORTLEVEL "ALL";
   DATAEXPORTOVERWRITEFILE ON;
};

FIX ("New York", "100-10");
   DATAEXPORT "File" "," "E:\temp\export1.txt";
ENDFIX;
```

You can declare an environment variable to reference the path and filename, `ENV_FILE="E:\temp\export1.txt"`, and use the following syntax in the calculation script:

```
DATAEXPORT "File" "," $ENV_FILE;
```

Essbase replaces the environment variable with the value taken from the user's environment.

In the following example, another environment variable is defined to export only Sales values (`CurrMbr="Sales"`):

```
SET DATAEXPORTOPTIONS
{
   DATAEXPORTLEVEL "ALL";
   DATAEXPORTOVERWRITEFILE ON;
};

FIX ("New York", "100-10", $CurrMbr);
   DATAEXPORT "File" "," $ENV_FILE;
ENDFIX;
```

Environment variables can also be used to parse arguments passed to RUNJAVA, an Essbase utility in which custom-defined functions can be called directly from a calculation script. For example, you can use environment variables to get user e-mail addresses. The following RUNJAVA statement sends an e-mail notification to explicitly-defined users (see the text in **bold**):

```
RUNJAVA com.hyperion.essbase.calculator.EssbaseAlert "localhost"
“to@somedomain.com” "cc@mydomain.com" "" "" "Mail Subject" "Mail Body" "";
```

You can declare environment variables for the users, `ENV_TOMAIL=“to@somedomain.com”` and `ENV_CCMAIL=“to@mydomain.com”`, and use the following syntax in the calculation script:

```
RUNJAVA com.hyperion.essbase.calculator.EssbaseAlert "localhost"
$ENV_TOMAIL $ENV_CCMAIL "" "" "Mail Subject" "Mail Body" "";
```

Using environment variables in calculation scripts is the same as using them in formulas. See "Using Environment Variables in Formulas" on page 368

Clearing Data

You can use the commands in Table 58 to clear data.

Table 58 List of Commands for Clearing Data

Calculation	Command
Changes the values of the cells you specify to #MISSING. The data blocks are not removed. You can use the FIX command with the CLEARDATA command to clear a subset of a database.	CLEARDATA
Remove the entire contents of a block, including all the dense dimension members. Essbase removes the entire block, unless CLEARBLOCK is inside a FIX command on members within the block.	CLEARBLOCK
Removes consolidated level blocks.	
Remove blocks containing derived values. Applies to blocks that are completely created by a calculation operation, not to blocks into which any values were loaded.	
Remove blocks for Dynamic Calc and Store member combinations. See Chapter 26, "Dynamically Calculating Data Values."	CLEARBLOCK DYNAMIC
Remove empty blocks	CLEARBLOCK EMPTY

If, in the Sample.Basic database, the Scenario dimension is dense, the following example removes all data cells that do not contain input data values and that intersect with member Actual from the Scenario dimension:

```
FIX(Actual)
   CLEARBLOCK NONINPUT;
ENDFIX
```

If the Scenario dimension is sparse, the following example removes only the blocks whose Scenario dimension member is Actual:

```
FIX(Actual)
   CLEARBLOCK NONINPUT;
ENDFIX
```

The following formula clears all the Actual data values for Colas:

```
CLEARDATA Actual -> Colas;
```

To clear an entire database, see "Clearing Data" in the *Oracle Essbase Administration Services Online Help*.

Copying Data

You can use the DATACOPY calculation command to copy data cells from one range of members to another range of members in a database. The two ranges must be the same size.

For example, in the Sample.Basic database, the following formula copies Actual values to Budget values:

```
DATACOPY Actual TO Budget;
```

You can use the FIX command to copy a subset of values.

For example, in the Sample.Basic database, the following formula copies Actual values to Budget values for the month of January only:

```
FIX (Jan)
   DATACOPY Actual TO Budget;
ENDFIX
```

See "Using the FIX Command" on page 460. For more information about the DATACOPY command, see the *Oracle Essbase Technical Reference*.

Calculating a Subset of a Database

➤ To calculate a subset of a database, use one of the following methods:

- Create a formula using member set functions to calculate lists of members.

  See "Calculating Lists of Members" on page 459.
- Use the FIX...ENDFIX commands to calculate a range of values by inclusion.

  See "Using the FIX Command" on page 460.
- Use the EXCLUDE...ENDEXCLUDE commands to calculate a range of values by exclusion.

  See "Using the Exclude Command" on page 461.

Note:

When Intelligent Calculation is turned on, the newly calculated data blocks are not marked as clean after a partial calculation of a database. When you calculate a subset of a database, you can use the SET CLEARUPDATESTATUS AFTER command to ensure that the newly calculated blocks are marked as clean. Using this command ensures that Essbase recalculates the database as efficiently as possible using Intelligent Calculation. See Chapter 25, "Understanding Intelligent Calculation."

Calculating Lists of Members

You can use a member set function to generate a list of members that is based on a member you specify. For example, the @IDESCENDANTS function generates a list of all the descendants of a specified member. When you use a member set function in a formula, Essbase generates a list of members before calculating the formula.

In the Sample.Basic database, the following example generates a list of these members—Total Expenses, Marketing, Payroll, and Misc:

```
@IDESCENDANTS("Total Expenses");
```

See the *Oracle Essbase Technical Reference*.

Using the FIX Command

Use the FIX command to define which members to include in the calculation.

The following examples are based on the Sample.Basic database.

- This example calculates only the Budget values for only the descendants of East (New York, Massachusetts, Florida, Connecticut, and New Hampshire):

  ```
  FIX(Budget,@DESCENDANTS(East))
     CALC DIM(Year, Measures, Product);
  ENDFIX
  ```

- This example fixes on member combinations for the children of East that have a UDA of New Mkt:

  ```
  FIX(@CHILDREN(East) AND @UDA(Market,"New Mkt"))
     Marketing = Marketing * 1.1;
  ENDFIX
  ```

  For information on defining UDAs, see Chapter 7, “Creating and Changing Database Outlines.”

- This example uses a wildcard match to fix on member names that end in the characters -10, which in Sample.Basic are members 100-10, 200-10, 300-10, and 400-10:

  ```
  FIX(@MATCH(Product, "???-10"))
     Price = Price * 1.1;
  ENDFIX
  ```

When you use the FIX command only on a dense dimension, Essbase retrieves the entire block that contains the required value or values for the members that you specify. I/O is not affected, and the calculation performance time is improved.

When you use the FIX command on a sparse dimension, Essbase retrieves the block for the specified sparse dimension members. I/O may be greatly reduced.

Essbase cycles through the database once for each FIX command that you use on dense dimension members. When possible, combine FIX blocks to improve calculation performance.

For example, by using one FIX command, the following calculation script causes Essbase to cycle through the database only once, calculating both the Actual and the Budget values:

```
FIX(Actual,Budget)
   CALC DIM(Year, Measures);
ENDFIX
```

In contrast, by using two FIX command, this calculation script causes Essbase to cycle through the database twice: once calculating the Actual data values; once calculating the Budget data values:

```
FIX(Actual)
   CALC DIM(Year, Measures);
ENDFIX
FIX(Budget)
   CALC DIM(Year, Measures);
ENDFIX
```

You cannot FIX on a subset of a dimension that you calculate within a FIX statement.

For example, the following calculation script returns an error message because the CALC DIM operation calculates the entire Market dimension, although the FIX above it fixes on specific members of the Market dimension:

```
FIX(@CHILDREN(East) AND @UDA(Market,"New Mkt"))
   CALC DIM(Year, Measures, Product, Market);
ENDFIX
```

Note:

The FIX command has restrictions. See the *Oracle Essbase Technical Reference*.

Using the Exclude Command

Use the EXCLUDE...ENDEXCLUDE command to define which members to exclude from the calculation. Sometimes it is easier to specify which members not to include in a calculation than to define which members to include.

Note:

The EXCLUDE command has some restrictions. See the *Oracle Essbase Technical Reference*.

Using DATAEXPORT to Export Data

The DATAEXPORT command enables calculation scripts to export data in binary or text, or directly to a relational database. A set of data-export-related calculation commands qualify what data to export and provide various output and formatting options.

The following command sequence shows the typical calculation script structure for exporting data:

```
SET DATAEXPORTOPTIONS
  {
    DATAEXPORTLEVEL parameters;
    DATAEXPORTDYNAMICCALC ON | OFF;
    DATAEXPORTNONEXISTINGBLOCKS ON | OFF;
    DATAEXPORTDECIMAL n;
    DATAEXPORTPRECISION n;
    DATAEXPORTCOLFORMAT ON | OFF;
    DATAEXPORTCOLHEADER dimensionName;
    DATAEXPORTDIMHEADER ON | OFF;
    DATAEXPORTRELATIONALFILE ON | OFF;
    DATAEXPORTOVERWRITEFILE ON | OFF;
    DATAEXPORTDRYRUN ON | OFF;
```

```
   };
DATAEXPORTCOND parameters;
FIX
  (fixMembers)
  DATAEXPORT parameters;
ENDFIX;
```

To develop a calculation script that exports a subset of data, first specify the SET DATAEXPORTOPTIONS command to define options for export content, format, and process (see Table 59).

Table 59 List of SET DATAEXPORTOPTIONS Commands

Calculation	Command
Content options	
Specify all, level 0, or input data value	DATAEXPORTLEVEL
Control export of dynamically calculated values	DATAEXPORTDYNAMICCALC
Specify whether to export data from all potential data blocks or only from existing data blocks.	DATAEXPORTNONEXISTINGBLOCKS
Specify the number of decimal positions in the exported values	DATAEXPORTDECIMAL
Specify the total number of positions in the exported values	DATAEXPORTPRECISION
Output format options	
Specify columnar or noncolumnar format	DATAEXPORTCOLFORMAT
Specify a dense dimension for the column header	DATAEXPORTCOLHEADER
Include a header record that lists all dimension names in the same order as the data in the file	DATAEXPORTDIMHEADER
Format the text export file for importing the data into a relational database	DATAEXPORTRELATIONALFILE
Processing options	
Specify whether an existing file with the same name and location is replaced	DATAEXPORTOVERWRITEFILE
Enable validating the set of calculation commands and viewing export statistics, including a time estimate—without having to perform the entire export process	DATAEXPORTDRYRUN

Options and parameters are optional, with default values. See the *Oracle Essbase Technical Reference*.

When exporting data to a binary or text file, you can specify a limit for the size of the export file with the EXPORTFILESIZELIMIT configuration setting in `essbase.cfg`. The minimum file size limit is 1 MB. The maximum size of the export file is limited only by factors such as file system limits (for example, some systems do not support files that are larger than 2 GB), available disk space, and user limits. By default, the maximum file size for the export file is 2 GB. If the exported data exceeds the file size limit set with EXPORTFILESIZELIMIT or, if using the default 2 GB limit on a file system that does not support large files, Essbase creates multiple export files, as needed. An underscore and number is appended to the names of the additional files, starting with _1. For example, if the filename is `outfile.txt` and three files are created, the resulting file names are `outfile.txt, outfile_1.txt,` and `outfile_2.txt`.

Use a DATAEXPORTCOND command to select data based on data values. Then use FIX...ENDFIX or EXCLUDE...ENDEXCLUDE calculation commands to select a slice of the database to be exported. Within the FIX...ENDFIX or EXCLUDE...ENDEXCLUDE group, include the DATAEXPORT command.

When using the DATAEXPORT command to export data for direct insertion into a relational database:

- The table to which the data is to be written must exist prior to data export
- Table and column names cannot contain spaces

By default, when inserting exported data, Essbase uses the row-insert method, in which each row is inserted one at a time. To improve performance, you can use the batch-insert method if your relational database and the ODBC driver support the functionality.

To enable batch insert, set the DATAEXPORTENABLEBATCHINSERT configuration setting in `essbase.cfg` to TRUE. To control the number of rows that are inserted at one time (instead of letting Essbase determine the batch size), use the DEXPSQLROWSIZE configuration setting to specify the number of rows in a batch (from 2 to 1000). If Essbase cannot determine whether the relational database and the ODBC driver support batch insert, it uses the row-insert method, and DEXPSQLROWSIZE (if set) is ignored.

Note:

If DATAEXPORTENABLEBATCHINSERT is set to TRUE and DEXPSQLROWSIZE is set to 1, batch insert is disabled (as a DEXPSQLROWSIZE setting of 1 inserts rows one at a time).

Use the DATAIMPORTBIN calculation command to import a previously exported binary export file. The SET DATAIMPORTIGNORETIMESTAMP calculation command enables you to manage the import requirement for a matching outline timestamp.

Other export methods include using ESSCMD, MaxL, and Administration Services Console for database backup. Report Writer can be used to select and format a database subset, creating an output text file (see "Exporting Text Data Using Report Scripts" on page 579).

Compared to using other methods to export data, using a calculation script has the following advantages and disadvantages:

- Advantages
  - Enables exporting a subset of data
  - Supports multiple targets: flat files, relational databases, and binary files
  - Provides options for type, format, or data
  - As part of a calculation script, can be deployed in a batch process
  - Can be very fast when the dynamic calculation export option is not used because DATAEXPORT directly accesses Kernel storage blocks in memory

- ❍ Provides, through binary export/import, a faster way to back up and restore data because the compressed format used by binary export requires less storage for the export files
- ❍ Can be used as a debug tool to trace batch calculation results by using the DATAEXPORT command before and after other calculation commands to track data changes

- Disadvantages
  - ❍ Contains limited data formatting options compared to Report Writer formatting
  - ❍ Works with stored members and Dynamic Calc members only, with no support for attribute members and alias names
  - ❍ Not supported for aggregate storage databases
  - ❍ Cannot export data directly to the client
  - ❍ Can significantly impact performance when exporting dynamic calculation data, unless you set DATAEXPORTNONEXISTINGBLOCKS to ON.

Enabling Calculations on Potential Blocks

When you use a formula on a dense member in a dense dimension, if the resultant values are from a dense dimension and the operand or operands are from a sparse dimension, Essbase does not automatically create the required blocks.

In the following example, based on Sample.Basic, assume that you want to create budget sales and expense data from existing actual data. Sales and Expenses are members in the dense Measures dimension; Budget and Actual are members in the sparse Scenario dimension.

```
FIX(Budget)
   (Sales = Sales -> Actual * 1.1;
   Expenses = Expenses -> Actual * .95;)
ENDFIX
```

Sales and Expenses, the results of the equations, are dense dimension members; the operand, Actual, is in a sparse dimension. Because Essbase executes dense member formulas only on existing data blocks, the calculation script does not create the required data blocks and Budget data values are not calculated for blocks that do not already exist.

You can solve the problem using the following techniques.

Using DATACOPY to Copy Existing Blocks

You can use the DATACOPY command to create a block for each existing block, and then perform calculations on the new blocks. For example:

```
DATACOPY Sales -> Actual TO Sales -> Budget;
DATACOPY Expenses -> Actual TO Expenses -> Budget;
FIX(Budget)
   (Sales = Sales -> Actual * 1.1;
   Expenses = Expenses -> Actual * .95;)
ENDFIX
```

Essbase creates blocks that contain the Budget values for each corresponding Actual block that exists. After the DATACOPY commands are finished, the remaining part of the script changes the values.

Using DATACOPY works well when:

- There is a mathematical relationship between values in existing blocks and their counterparts created by the DATACOPY.

  For example, in the preceding example, the Budget values can be calculated based on the existing Actual values.

  Caution!

  DATACOPY creates the new blocks with identical values in all cells from the source blocks. If the formula performs only on a portion of the block, these copied cells remain at the end of calculation, potentially resulting in unwanted values.

- None of the blocks that are copied contain only #MISSING values.

  If #MISSING values exist, blocks are written that contain only #MISSING values. Unneeded #MISSING blocks require Essbase resource and processing time.

Using SET CREATENONMISSINGBLK to Calculate All Potential Blocks

If you are concerned about unwanted values, instead of using DATACOPY, you can use the SET CREATENONMISSINGBLK ON calculation command, which calculates all potential blocks in memory and then stores only the calculated blocks that contain data values. The SET CREATENONMISSINGBLK calculation command can be useful when calculating values on dense or sparse dimensions.

The following example creates budget sales and expense data from existing actual data. Sales and Expenses are members in the dense Measures dimension; Budget and Actual are members in the sparse Scenario dimension.

```
FIX(Budget)
SET CREATENONMISSINGBLK ON
   (Sales = Sales -> Actual * 1.1;
   Expenses = Expenses -> Actual * .95;)
ENDFIX
```

Note:

If SET CREATEBLOCKONEQ ON is set for sparse dimensions, SET CREATENONMISSINGBLK ON temporarily overrides it until a SET CREATENONMISSINGBLK OFF command is encountered or the calculation script is completed. See "Nonconstant Values Assigned to Members in a Sparse Dimension" on page 877.

The advantage of using the SET CREATENONMISSINGBLK command is that, when applied on dense members, only data cells that are affected by the member formula are saved. The disadvantage is that too many potential blocks may be materialized in memory, possibly affecting calculation performance. When you use this command, limit the number of potential blocks; for example, by using FIX to restrict the scope of the blocks to be calculated.

See the *Oracle Essbase Technical Reference*.

Writing Calculation Scripts for Partitions

A partitioned application can span multiple servers, processors, or computers.

You can achieve significant calculation performance improvements by partitioning applications and running separate calculations on each partition. When using partitioning:

- Evaluate the performance impact on the overall database calculation. To improve performance, you can:
  - Redesign the overall calculation to avoid referencing remote values that are in a transparent partition in a remote database
  - Dynamically calculate a value in a remote database.

    See "Dynamically Calculating Data in Partitions" on page 433.
  - Replicate a value in the database that contains the applicable formula.

    For example, if replicating quarterly data for the Eastern region, replicate only the values for Qtr1, Qtr2, Qtr3, and Qtr4, and calculate the parent Year values locally.
- Ensure that a referenced value is up-to-date when Essbase retrieves it. Choose one of the options previously discussed (redesign, dynamically calculate, or replicate) or calculate the referenced database before calculating the formula.

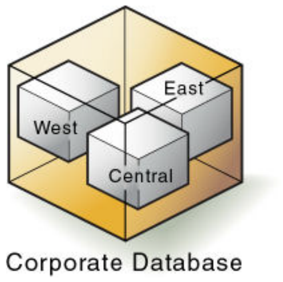
See Chapter 14, "Designing Partitioned Applications" and Chapter 15, "Creating and Maintaining Partitions."

Controlling Calculation Order for Partitions

You must calculate databases in a specific order to ensure that Essbase calculates the required results.

The example in Figure 136 shows partitions in which you view information from the West, Central, and East databases transparently from the Corporate database.

Figure 136 Calculating Partitions



West, Central, and East contain only actual values. Corporate contains actual and budgeted values. Although you can view West, Central, and East data in the Corporate database, the data exists only in the West, Central, and East databases—it is not duplicated in the Corporate database.

Therefore, when Essbase calculates Corporate, it must take the latest values from West, Central, and East. To obtain the required results, you must calculate West, Central, and East before you calculate Corporate.

Reviewing the Process for Creating Calculation Scripts

Use this process to create a calculation script:

1. Create a calculation script or open an existing calculation script.

   See "Creating Scripts" or "Opening Scripts" in the *Oracle Essbase Administration Services Online Help*.

2. Enter or edit the contents of the calculation scripts.

   See "About Calculation Script Editor" in the *Oracle Essbase Administration Services Online Help* for information about:

   - Associating a script with an outline
   - Searching an outline tree for members
   - Inserting dimensions, members, and aliases in a script from an outline tree
   - Inserting functions and commands in a script from a tree
   - Using syntax autocompletion
   - Checking script syntax
   - Executing scripts
   - Viewing color-coded script elements
   - Searching for text in a script
   - Changing fonts

3. Validate the calculation script.

   See "Checking Syntax" on page 468 and "Checking Script Syntax" in the *Oracle Essbase Administration Services Online Help*.

4. Save the calculation script.

   See "Saving Calculation Scripts" on page 468 and "Saving Scripts" in the *Oracle Essbase Administration Services Online Help*.

5. Execute the calculation script.

   See "Executing Calculation Scripts" on page 469, "Checking the Results of Calculations" on page 469, and "Executing Calculation Scripts" in the *Oracle Essbase Administration Services Online Help*.

6. If necessary, perform other operations on the calculation script.

In the *Oracle Essbase Administration Services Online Help*, see the following topics:

- “Locking and Unlocking Objects”
- “Copying Scripts”
- “Renaming Scripts”
- “Deleting Scripts”
- “Printing Scripts”

Checking Syntax

Essbase includes a syntax checker that flags syntax errors (such as a mistyped function name) in a calculation script. The results are displayed in the messages panel in Administration Services Console.

If syntax errors are not found, Essbase indicates the syntax check succeeded.

If syntax errors are found, Essbase indicates the syntax check failed, and displays one error at a time. Typically, an error message includes the line number in which the error occurred and a brief description. For example, if a semicolon end-of-line character is missing at the end of a calculation script command, Essbase displays a message similar to this one:

```
Error: line 1: invalid statement; expected semicolon
```

When you reach the last error, Essbase displays the following message:

```
No more errors
```

➤ To check the syntax of a calculation script in Calculation Script Editor, see “Checking Script Syntax” in the *Oracle Essbase Administration Services Online Help*.

Note:

The syntax checker cannot determine semantic errors, which occur when a calculation script does not work as you expect. To find semantic errors, run the calculation and check the results to ensure they are as you expect.

Saving Calculation Scripts

You can save a calculation script in the following locations:

- As a file on a client computer.
- As an artifact on an Essbase Server, which allows other users to access the calculation script. You can associate the script with the following artifacts:
  - An application and all the databases within the application, which lets you run the script against any database in the application.

    Calculation scripts associated with an application are saved in the `*ARBORPATH*/app/*appname*` directory on the Essbase Server computer.

- A database, which lets you run the script against the specified database.

  Calculation scripts associated with a database are saved in the `ARBORPATH/app/appname/dbname` directory on the Essbase Server computer.

➤ To save a calculation script using Calculation Script Editor, see "Saving Scripts" in the *Oracle Essbase Administration Services Online Help*.

Executing Calculation Scripts

Before you can execute a calculation script in Administration Services, you must save it as an artifact on an Essbase Server, a client computer, or a network. See "Saving Calculation Scripts" on page 468.

When you use Administration Services to execute a calculation script, you can execute the calculation in the background so that you can continue working as the calculation processes. You can then check the status of the background process to see when the calculation has completed. See "Executing Calculation Scripts" in the *Oracle Essbase Administration Services Online Help*.

➤ To execute a calculation script, use a tool:

Tool	Topic	Location
Administration Services	Executing Calculation Scripts	*Oracle Essbase Administration Services Online Help*
MaxL	**execute calculation**	*Oracle Essbase Technical Reference*
ESSCMD	RUNCALC	*Oracle Essbase Technical Reference*
Spreadsheet Add-in	ESSBASE, then CALCULATION	*Oracle Essbase Spreadsheet Add-in User's Guide*
Smart View	Calculating Data	*Oracle Hyperion Smart View for Office Online Help*

Checking the Results of Calculations

After you execute a calculation script, you can check the results of the calculation in Smart View or Spreadsheet Add-in.

Essbase provides the following information about the executed calculation script:

- Calculation order of the dimensions for each pass through the database
- Total calculation time

To display more-detailed information, you can use the SET MSG SUMMARY, SET MSG DETAIL, and SET NOTICE commands in a calculation script. See "Specifying Global Settings for a Database Calculation" on page 450.

You can use these messages to understand how the calculation is performed and to tune it for the next calculation.

Where you view this information depends on the tool used to execute the calculation script.

- Administration Services, Spreadsheet Add-in, and Smart View: Application log

  See "Viewing the Essbase Server and Application Logs" on page 742.
- MaxL: Standard output (command-line window)

  The amount of information depends on the message level set in MaxL Shell.
- ESSCMD: ESSCMD window or standard output (command-line window)

  The amount of information depends on the message level set in ESSCMD.

Copying Calculation Scripts

You can copy calculation scripts to applications and databases on any Essbase Server, according to your permissions. You can also copy scripts across servers as part of application migration.

➤ To copy a calculation script, use a tool:

Tool	Topic	Location
Administration Services	Copying Scripts	*Oracle Essbase Administration Services Online Help*
MaxL	**create calculation as**	*Oracle Essbase Technical Reference*
ESSCMD	COPYOBJECT	*Oracle Essbase Technical Reference*

29 Reviewing Examples of Calculation Scripts

In This Chapter

The information in this chapter applies only to block storage databases and is not relevant to aggregate storage databases. Also see Chapter 57, "Comparison of Aggregate and Block Storage."

Introduction

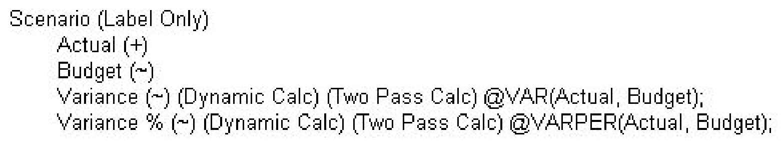
All examples in this chapter are based on the Sample.Basic database.

For examples that use the Intelligent Calculation commands SET UPDATECALC and SET CLEARUPDATESTATUS, see "Reviewing Examples That Use SET CLEARUPDATESTATUS" on page 408 and "Reviewing Examples and Solutions for Multiple-Pass Calculations" on page 412.

Calculating Variance

This example includes a calculation of the variance percentage between Budget and Actual values.

Figure 137 Calculating Variance and Variance %



```
Scenario (Label Only)
      Actual (+)
      Budget (~)
      Variance (~) (Dynamic Calc) (Two Pass Calc) @VAR(Actual, Budget);
      Variance % (~) (Dynamic Calc) (Two Pass Calc) @VARPER(Actual, Budget);
```

During a default calculation, Essbase aggregates the values on the Market and Product dimensions. Because percentage values do not aggregate correctly, the Variance % formula must be recalculated after the default calculation.

In the Sample.Basic outline, Variance % is tagged as a Dynamic Calc, two-pass member. Therefore, Essbase dynamically calculates Variance % values when they are retrieved. The dynamic calculation overwrites the incorrect values with the correctly calculated percentages.

If you choose not to tag Variance % as a Dynamic Calc, two-pass member, use the following calculation script—which assumes that Intelligent Calculation is turned on (the default)—to perform a default calculation and to recalculate the formula on Variance %:

```
CALC ALL;
SET UPDATECALC OFF;
SET CLEARUPDATESTATUS AFTER;
"Variance %";
```

Essbase performs the following actions:

1. Performs a default calculation of the database (CALC ALL).

   Alternatively, you can run a default calculation of the database outline without using a calculation script.

2. Turns off Intelligent Calculation (SET UPDATECALC OFF).
3. Marks the calculated blocks calculated by the variance formula of the calculation script as clean, even though the variance calculation is a partial calculation of the database (CLEARUPDATESTATUS AFTER).

   By default, data blocks are marked as clean only after a full calculation of the database.

4. Cycles through the database calculating the formula for Variance %.

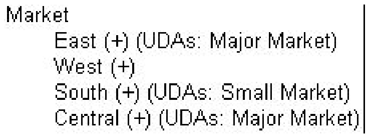
See "Choosing Two-Pass Calculation Tag or Calculation Script" on page 886 and "Using Two-Pass Calculation" on page 883.

For information on calculating *statistical* variance, see the *Oracle Essbase Technical Reference*.

Calculating Database Subsets

This example shows how a regional Marketing manager can calculate her respective area of the database. The calculation script uses @DESCENDENTS(East) to limit the calculations to the East region, as it calculates the Year, Measures, and Product dimensions for each child of East.

Figure 138 Market Dimension from the Sample.Basic Database



```
Market
    East (+) (UDAs: Major Market)
    West (+)
    South (+) (UDAs: Small Market)
    Central (+) (UDAs: Major Market)
```

```
/* Calculate the Budget data values for the descendants of East */

FIX(Budget, @DESCENDANTS(East))
   CALC DIM(Year, Measures, Product);
ENDFIX

/* Consolidate East */

FIX(Budget)
   @DESCENDANTS(East);
ENDFIX
```

Essbase performs the following actions:

1. Fixes on the Budget values of the descendants of East.
2. Calculates the Year, Measures, and Product dimensions in one pass of the database for all Budget values of the descendants of East.
3. Fixes on the Budget values for all members on the other dimensions.
4. Aggregates the descendants of East and places the result in East.

Loading New Budget Values

This example calculates Budget values and then recalculates the Variance and Variance % members:

```
/* Calculate all Budget values */

FIX(Budget)
   CALC DIM(Year, Product, Market, Measures);
ENDFIX

/* Recalculate the Variance and Variance % formulas,
   which requires two passes */

Variance;
"Variance %";
```

Essbase performs the following actions:

1. Fixes on the Budget values.
2. Calculates all Budget values.

   The CALC DIM command is used to calculate all the dimensions except for the Scenario dimension, which contains Budget.
3. Calculates the formula applied to Variance in the database outline.

4. Calculates the formula applied to Variance % in the database outline.

Calculating Product Share and Market Share Values

This example calculates product share and market share values for each market and each product. The share values are calculated as follows:

- Each member as a percentage of the total
- Each member as a percentage of its parent

Assume that you added four members to the Measures dimension:

- Market Share
- Product Share
- Market %
- Product %

```
/* First consolidate the Sales values to ensure that they are accurate */

FIX(Sales)
   CALC DIM(Year, Market, Product);
ENDFIX

/* Calculate each market as a percentage of the
   total market for each product */

"Market Share" = Sales % Sales -> Market;

/* Calculate each product as a percentage of the
   total product for each market */

"Product Share" = Sales % Sales -> Product;

/* Calculate each market as a percentage of its
   parent for each product */

"Market %" = Sales % @PARENTVAL(Market, Sales);

/* Calculate each product as a percentage its
   parent for each market */

"Product %" = Sales % @PARENTVAL(Product, Sales);
```

Essbase performs the following actions:

1. Fixes on the Sales values and consolidates all the Sales values.

   The CALC DIM command is used to calculate the Year, Market, and Product dimensions. The Measures dimension contains the Sales member and therefore is not consolidated. The Scenario dimension is label only and therefore does not need to be consolidated.
2. Cycles through the database and calculates Market Share by taking the Sales value for each product in each market for each month and calculating this Sales value as a percentage of total Sales in all markets for each product (Sales -> Market).

3. Calculates Product Share by taking the Sales value for each product in each market for each month and calculating this Sales value as a percentage of total Sales of all products in each market (Sales -> Product).
4. Calculates Market % by taking the Sales value for each product in each market for each month and calculating this Sales value as a percentage of the Sales value of the parent of the current member on the Market dimension.

   The @PARENTVAL function is used to obtain the Sales value of the parent on the Market dimension.
5. Calculates Market % by taking the Sales value for each product in each market for each month, and calculating this Sales value as a percentage of the Sales value of the parent of the current member on the Product dimension.

   The @PARENTVAL function is used to obtain the Sales value of the parent on the Product dimension.

Allocating Costs Across Products

This example allocates overhead costs to each product in each market for each month. Overhead costs are allocated based on each product's Sales value as a percentage of the total Sales for all products.

Assume that you added two members to the Measures dimension:

- OH_Costs for the allocated overhead costs
- OH_TotalCost for the total overhead costs

```
/* Declare a temporary array called ALLOCQ
   based on the Year dimension */

ARRAY ALLOCQ[Year];

/* Turn the Aggregate Missing Values setting off.
   If this is your system default, omit this line */

SET AGGMISSG OFF;

/* Allocate the overhead costs for Actual values */

FIX(Actual)
   OH_Costs (ALLOCQ=Sales/Sales->Product; OH_Costs =
   OH_TotalCost->Product * ALLOCQ;);

/* Calculate and consolidate the Measures dimension */

   CALC DIM(Measures);
ENDFIX
```

Essbase performs these calculations:

1. Creates a one-dimensional array called ALLOCQ to store the value of Sales as a percentage of total Sales temporarily for each member combination.

The size of ALLOCQ is based on the number of members in the Year dimension.

2. #MISSING values are not aggregated to their parents (SET AGGMISSG OFF). Data values stored at parent levels are not overwritten.

   If SET AGGMISSG OFF is your system default, omit this line. See "Consolidating #MISSING Values" on page 894.

   - Fixes on the Actual values.
   - Cycles through the member combinations for Actual and calculates OH_Costs.
   - Takes the Sales value for each product in each market for each month and calculates it as a percentage of total Sales for all products in each market (Sales -> Product). The result is placed in ALLOCQ.
   - Takes the total overhead costs for all products (OH_TotalCost -> Product) and multiplies it by the value it has just placed in ALLOCQ. The result is placed in OH_Costs.

     Note that both equations are enclosed in parentheses ( ) and associated with the OH_Costs member: OH_Costs (equation1; equation2;).

3. Calculates and consolidates the Measures dimension.

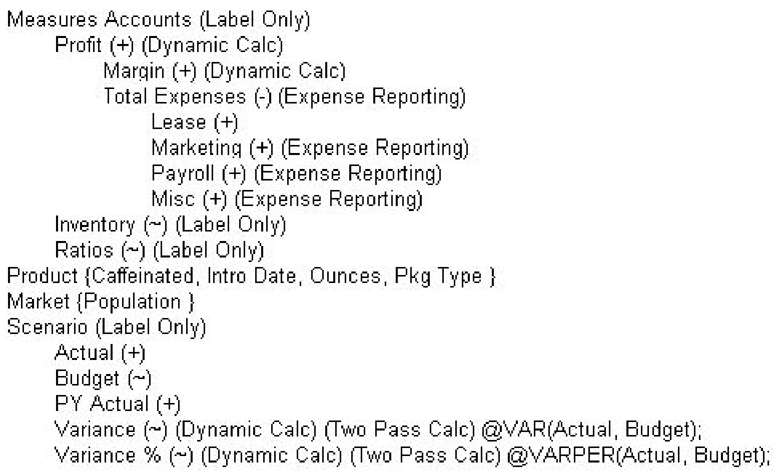
Allocating Values Within a Dimension

This example uses the @ALLOCATE function to allocate budgeted total expenses across expense categories for two products. The budgeted total expenses are allocated based on the actual values for the previous year.

Assume that you made the following changes:

- Added a child, Lease, under Total Expenses in the Measures dimension
- Added a child, PY Actual, to the Scenario dimension
- Removed the Dynamic Calc tag from the Total Expenses member

Figure 139 Modified Measures and Scenario Dimensions from the Sample.Basic Database



```
Measures Accounts (Label Only)
    Profit (+) (Dynamic Calc)
        Margin (+) (Dynamic Calc)
        Total Expenses (-) (Expense Reporting)
            Lease (+)
            Marketing (+) (Expense Reporting)
            Payroll (+) (Expense Reporting)
            Misc (+) (Expense Reporting)
    Inventory (~) (Label Only)
    Ratios (~) (Label Only)
Product {Caffeinated, Intro Date, Ounces, Pkg Type }
Market {Population }
Scenario (Label Only)
    Actual (+)
    Budget (~)
    PY Actual (+)
    Variance (~) (Dynamic Calc) (Two Pass Calc) @VAR(Actual, Budget);
    Variance % (~) (Dynamic Calc) (Two Pass Calc) @VARPER(Actual, Budget);
```

Assume that data values of 1000 and 2000 are loaded into Budget -> Total Expenses for Colas and Root Beer, respectively. These values must be allocated to each expense category, evenly spreading the values based on the nonmissing children of Total Expenses from PY Actual. The allocated values must be rounded to the nearest dollar.

```
/* Allocate budgeted total expenses based on prior year */

FIX("Total Expenses")
   Budget = @ALLOCATE(Budget->"Total Expenses",
   @CHILDREN("Total Expenses"),"PY Actual",,
   spread,SKIPMISSING,roundAmt,0,errorsToHigh)
ENDFIX
```

The following table shows the results of the calculation script:

		Budget	**PY Actual**
Colas	Marketing	334[*]	150
	Payroll	#MI	#MI
	Lease	333	200
	Misc	333	100
	Total Expenses	**1000**	**450**
Root Beer	Marketing	500	300
	Payroll	500	200
	Lease	500	200
	Misc	500	400
	Total Expenses	**2000**	**1100**

[*]Rounding errors are added to this value. See step 5 on page 477.

➤ Essbase cycles through the database, performing the following calculations:

1. **Fixes on the children of Total Expenses.**

   Using a FIX statement with @ALLOCATE may improve calculation performance.

2. **For Budget -> Colas -> Marketing, divides 1 by the count of nonmissing values for each expense category in PY Actual -> Colas for each month.**

   In this case, 1 is divided by 3, because there are 3 nonmissing expense values for Budget -> Colas.

3. **Takes the value from step 2 (.333), multiplies it by the value for Budget -> Colas -> Total Expenses (1000), and rounds to the nearest dollar (333). The result is placed in Budget -> Colas -> Marketing.**

4. **Repeats step 2 and step 3 for each expense category for Budget -> Colas and then for Budget -> Root Beer.**

5. **As specified in the calculation script, rounds allocated values to the nearest whole dollar.**

   Essbase makes a second pass through the block to make the sum of the rounded values equal to the allocation value (for example, 1000 for Budget -> Colas -> Total Expenses). In this example, there is a rounding error of 1 for Budget -> Colas -> Total Expenses, because the expense categories add up to 999, not 1000, which is the allocation value. Because all allocated values are identical (333), the rounding error of 1 is added to the first value in the allocation range, Budget -> Colas -> Marketing (thus a value of 334).

Allocating Values Across Multiple Dimensions

This example uses the @MDALLOCATE function to allocate a loaded value for budgeted total expenses across three dimensions. The budgeted total expenses are allocated based on the actual values of the previous year.

Assume that you made the following changes:

- Added a child, PY Actual, to the Scenario dimension
- Copied data from Actual into PY Actual
- Cleared data from Budget

For this example, a value of 750 (for Budget -> Total Expenses -> Product -> East -> Jan) must be allocated to each expense category for the children of product 100 across the states in the East. The allocation uses values from PY Actual to determine the percentage share that each category should receive.

This calculation script defines the allocation:

```
/* Allocate budgeted total expenses based on prior year,
   across 3 dimensions */

SET UPDATECALC OFF;
FIX (East, “100”, “Total Expenses”)
   BUDGET = @MDALLOCATE(750,3,@CHILDREN(“100”),@CHILDREN("Total
Expenses"),@CHILDREN(East),"PY Actual",,share);
ENDFIX
```

The following table shows the values for PY Actual:

		Jan			
		PY Actual			
		Marketing	**Payroll**	**Misc**	**Total Expenses**
100-10	New York	94	51	0	145
	Massachusetts	23	31	1	55
	Florida	27	31	0	58
	Connecticut	40	31	0	71
	New Hampshire	15	31	1	47
100-20	New York	199	175	2	376

		Jan			
		PY Actual			
		Marketing	**Payroll**	**Misc**	**Total Expenses**
	Massachusetts	#MI	#MI	#MI	#MI
	Florida	#MI	#MI	#MI	#MI
	Connecticut	26	23	0	49
	New Hampshire	#MI	#MI	#MI	#MI
100-30	New York	#MI	#MI	#MI	#MI
	Massachusetts	26	23	0	49
	Florida	#MI	#MI	#MI	#MI
	Connecticut	#MI	#MI	#MI	#MI
	New Hampshire	#MI	#MI	#MI	#MI
100	New York	#MI	#MI	#MI	#MI
	Massachusetts	12	22	1	35
	Florida	12	22	1	35
	Connecticut	94	51	0	145
	New Hampshire	23	31	1	55
	East	237	220	3	460

➤ Essbase cycles through the database, performing these calculations:

1 Fixes on East, the children of 100, and Total Expenses.

Using a FIX statement with @MDALLOCATE may improve calculation performance.

2 Before performing the allocation, determines what share of 750 (the value to be allocated) each expense category should receive, for each product-state combination, using the shares of each expense category from PY Actual. Starting with PY Actual -> 100-10 -> New York, Essbase divides the value for the first expense category, Marketing, by the value for PY Actual-> 100-10 -> East -> Total Expenses to calculate the percentage share of that category.

For example, Essbase divides the value for PY Actual -> 100-10 -> New York -> Marketing (94) by the value for PY Actual -> 100-10 -> East -> Total Expenses (460), which yields a percentage share of approximately 20.4% for the Marketing category.

3 Repeats step 2 for each expense category, for each product-state combination.

4 During the allocation, Essbase uses the percentage shares calculated in step 2 and step 3 to determine what share of 750 should be allocated to each child of Total Expenses from Budget, for each product-state combination.

For example, for Marketing, Essbase uses the 20.4% figure calculated in step step 2, takes 20.4% of 750 (approximately 153), and places the allocated value in Budget -> 100-10 -> New York -> Marketing (see the table that follows this procedure).

5 **Repeats step 4 for each expense category and for each product-state combination, using the percentage shares from PY Actual calculated in step 2 and step 3.**

6 **Consolidates the expense categories to yield the values for Total Expenses.**

The following table shows the results of the allocation for Budget:

		Jan			
		Budget			
		Marketing	**Payroll**	**Misc**	**Total Expenses**
100–10	New York	153.26	83.15	0	236.41
	Massachusetts	37.50	50.54	1.63	89.67
	Florida	44.02	50.54	0	94.56
	Connecticut	65.22	50.54	0	115.76
	New Hampshire	24.46	50.54	1.63	76.63
100–20	New York	#MI	#MI	#MI	#MI
	Massachusetts	#MI	#MI	#MI	#MI
	Florida	42.39	37.50	0	79.89
	Connecticut	#MI	#MI	#MI	#MI
	New Hampshire	#MI	#MI	#MI	#MI
100–30	New York	#MI	#MI	#MI	#MI
	Massachusetts	#MI	#MI	#MI	#MI
	Florida	#MI	#MI	#MI	#MI
	Connecticut	#MI	#MI	#MI	#MI
	New Hampshire	19.57	35.87	1.63	57.07
100	New York	153.26	83.15	0	236.41
	Massachusetts	37.50	50.54	1.63	89.67
	Florida	86.41	88.04	0	174.46
	Connecticut	65.22	50.54	0	115.76
	New Hampshire	44.02	86.41	3.26	133.70
	East	386.41	358.70	4.89	750

Goal-Seeking Using the LOOP Command

This example shows how to calculate the sales value you must reach to obtain a certain profit on a specific product. In this case, the calculation script adjusts the Budget value of Sales to reach a goal of 15,000 Profit for Jan.

Assume that no members are tagged as Dynamic Calc, and that the Profit per Ounce member (under Ratios in the Scenario dimension) is not included in the calculation.

Figure 140 Measures Dimension from the Sample.Basic Database

```
Measures Accounts (Label Only)
    Profit (+)
        Margin (+)
            Sales (+)
            COGS (-) (Expense Reporting)
        Total Expenses (-) (Expense Reporting)
            Marketing (+) (Expense Reporting)
            Payroll (+) (Expense Reporting)
            Misc (+) (Expense Reporting)
    Inventory (~) (Label Only)
    Ratios (~) (Label Only)
        Margin % (+) (Two Pass Calc) Margin % Sales;
        Profit % (~) (Two Pass Calc) Profit % Sales;
```

Assume that, before running the goal-seeking calculation script, the data values are:

Product, Market, Budget	Jan
Profit	12,278.50
Margin	30,195.50
Sales	49,950.00
COGS	19,755.00
Total Expenses	17,917.00
Marketing	3,515.00
Payroll	14,402.00
Misc	0
Inventory	Label Only member
Ratios	Label Only member
Margin %	60.45
Profit %	24.58

```
/* Declare the temporary variables and set their initial values*/

VAR
   Target = 15000,
   AcceptableErrorPercent = .001,
   AcceptableError,
   PriorVar,
   PriorTar,
   PctNewVarChange = .10,
   CurTarDiff,
   Slope,
   Quit = 0,
      DependencyCheck,
      NxtVar;

/*Declare a temporary array variable called Rollback
   based on the Measures dimension */

ARRAY Rollback [Measures];

/* Fix on the appropriate member combinations and perform the
   goal-seeking calculation*/

FIX(Budget, Jan, Product, Market)
   LOOP (35, Quit)
      Sales (Rollback = Budget;
      AcceptableError = Target * (AcceptableErrorPercent);
      PriorVar = Sales;
      PriorTar = Profit;
      Sales = Sales + PctNewVarChange * Sales;);
      CALC DIM(Measures);
      Sales (DependencyCheck = PriorVar - PriorTar;
      IF(DependencyCheck <> 0) CurTarDiff = Profit - Target;
         IF(@ABS(CurTarDiff) > @ABS(AcceptableError))
            Slope = (Profit - PriorTar) / (Sales - PriorVar);
            NxtVar = Sales - (CurTarDiff / Slope);
            PctNewVarChange = (NxtVar - Sales) / Sales;
         ELSE
            Quit = 1;
         ENDIF;
      ELSE
         Budget = Rollback;
         Quit = 1;
      ENDIF;);
   ENDLOOP
   CALC DIM(Measures);
ENDFIX
```

➤ Essbase performs the following calculations:

1. **Declares the required temporary variables using the VAR command. Where appropriate, the initial values are set.**
2. **Declares a one-dimensional array called Rollback to store the Budget values.**

   The size of Rollback is based on the number of members in the Measures dimension.
3. **Fixes on the Jan -> Budget values for all Product and Market members.**
4. **Ensures that the commands between LOOP and ENDLOOP are cycled through 35 times *for each member combination*. If, however, the Quit variable is set to 1, the LOOP is broken and the calculation continues after the ENDLOOP command.**
5. **Cycles through the member combinations, performing the following calculations:**

   a. Places the Budget -> Sales value in the Rollback temporary array variable.

b. Calculates the acceptable error, by multiplying the Target value (15000) by the AcceptableErrorPercent value (0.001). The result is placed in the AcceptableError variable.

c. Retains the current Sales value, and places the Sales value for the current member combination in the PriorVar temporary variable.

d. Retains the current Profit value, and places the Profit value for the current member combination in the PriorTar temporary variable.

e. Calculates a new Sales value by multiplying the PctNewVarChange value (0.1) by the current Sales value, and adding the current Sales value. The result is placed in Sales.

f. Calculates and consolidates the Measures dimension.

g. Subtracts the PriorTar value from the PriorVar value, and places the result in the DependencyCheck temporary variable.

h. Checks that DependencyCheck is not 0 (zero) (IF).

- ❍ If DependencyCheck is not 0, subtracts the Target value (15000) from the current Profit and places the result in the CurTarDiff temporary variable.

  The IF command checks whether the absolute value (irrespective of the + or – sign) of CurTarDiff is greater than the absolute value of AcceptableError:

  - ❑ If greater than AcceptableError, calculates the Slope, NxtVar, and PctNewVarChange temporary variables.
  - ❑ If not greater than AcceptableError, breaks the LOOP command by setting the value of Quit to 1. The calculation continues after the ENDLOOP command.

- ❍ If DependencyCheck is 0, places the value in the Rollback array into Budget. Essbase breaks the LOOP command by setting the value of Quit to 1. The calculation continues after the ENDLOOP command.

6 **Calculates and consolidates the Measures dimension.**

The following table shows the results for product 100-10:

Product, Market, Budget	Jan
Profit	15,000.00
Margin	32,917.00
Sales	52,671.50
COGS	19,755.00
Total Expenses	17,917.00
Marketing	3,515.00
Payroll	14,402.00
Misc	0
Inventory	Label Only member
Ratios	Label Only member

Product, Market, Budget	Jan
Margin %	28.47839913
Profit %	62.49489762

Forecasting Future Values

This example uses a linear regression forecasting method to produce a trend (@TREND), or line, that starts with the known data values from selected previous months and continues with forecasted values based on the known values, and shows how to check the results of the trend for "goodness of fit" to the known data values. In this case, the calculation script forecasts sales data for June–December, assuming that data currently exists only up to May.

Assume that the Measures dimension contains an additional child, ErrorLR, where the goodness-of-fit results are placed.

```
Sales
(@TREND(@LIST(Jan,Mar,Apr),@LIST(1,3,4),,
  @RANGE(ErrorLR,@LIST(Jan,Mar,Apr)),
    @LIST(6,7,8,9,10,11,12),
      Jun:Dec,LR););
```

The following table describes the parameters used in the calculation script:

Parameter	Description
@LIST(Jan,Mar,Apr)	Represents the *Ylist*, or the members that contain the known data values. The @LIST function groups the three members as a comma-delimited list and keeps the list separate from other parameters.
@LIST(1,3,4)	Represents the *Xlist*, or the underlying variable values. Because Feb and May are skipped, Essbase numbers the *Ylist* values as 1,3,4.
,	The extra comma after the *Xlist* parameter indicates that a parameter (*weightList*) was skipped. This example uses a default weight of 1.
@RANGE(ErrorLR,@LIST(Jan,Mar,Apr)	Represents the *errorList*, or the member list where results of the goodness of fit of the trend line to *Ylist* are placed. The values placed in *errorList* are the differences between the data points in *Ylist* and the data points on the trend line that is produced. The @RANGE function combines the ErrorLR member with *Ylist* (Jan, Mar, Apr) to produce a member list.
@LIST(6,7,8,9,10,11,12)	Represents the *XforecastList*, or the underlying variable values for which the forecast is sought. This example forecasts values consecutively for Jun–Dec, so the values are 6,7,8,9,10,11,12.
Jun:Dec	Represents the *YforecastList*, or the member list into which the forecast values are placed. This example forecasts values for Jun–Dec, based on the values for Jan, Mar, and Apr.

Parameter	Description
LR	Specifies the Linear Regression method.

➤ Essbase cycles through the database, performing the following calculations:

1. **Finds the known data values on which to base the trend (Sales for Jan, Mar, Apr), as specified by the *`Ylist`* and *`Xlist`* parameters.**
2. **Calculates the trend line using Linear Regression and places the results in Sales for Jun–Dec, as specified by the *`YforecastList`* parameter.**
3. **Calculates the goodness of fit of the trend line for the data values for Jan, Mar, and Apr, and places the results in ErrorLR for those months.**

   For example, the value in ErrorLR for Jan (4.57) means that after Essbase calculates the trend line, the difference between the Sales value for Jan (2339) and the Jan value on the trend line is 4.57. The ErrorLR values for Feb and May are #MISSING, because these months were not part of *`Ylist`*.

   The following table shows the results of the calculation script:

	100 West Actual	
	Sales	ErrorLR
Jan	2339	4.57
Feb	2298	#MI
Mar	2313	-13.71
Apr	2332	9.14
May	2351	#MI
Jun	2315.14	#MI
Jul	2311.29	#MI
Aug	2307.49	#MI
Sep	2303.57	#MI
Oct	2299.71	#MI
Nov	2295.86	#MI
Dec	2292	#MI

30 Developing Custom-Defined Calculation Macros

In This Chapter

The information in this chapter applies only to block storage databases and is not relevant to aggregate storage databases. Also see Chapter 57, "Comparison of Aggregate and Block Storage."

Understanding Custom-Defined Macros

With *custom-defined macros*, you can combine multiple calculation functions into a single function.

When developing and testing custom-defined macros, create and test new macros locally within a test application. Register custom-defined macros globally only after you have tested them in a test application and are ready to use them in a production environment.

To create and manage custom-defined macros, you must have a security level of Database Manager or higher.

Naming Custom-Defined Macros

Follow these guidelines when naming custom-defined macros:

- Start the macro name with the "@" symbol; for example, @MYMACRO. The rest of a name can contain letters, numbers, and the following symbols: @, #, $, and _. Macro names must not contain spaces.
- For macros that are called only by other macros, start the macro name with "@_", to distinguish it from general-use macros and functions.

- Give macros unique names. Additionally, a macro name must be different from the names of custom-defined functions and from the names of existing calculation functions.

  Note:

  If an application contains a local macro that has the same name as a global macro, the local macro takes precedence and is used for calculation.

- For local macros, you must prepend the application name to the macro name, separating the application name from the macro name with a period:

  `AppName.@MacroName`

  For example:

  `Sample.@MYMACRO`

- Because global macros are available to any application running on the Essbase Server where the macro was created, you do not assign an application name to it.

Creating Custom-Defined Macros

When you create a custom-defined macro, Essbase records the macro definition and stores it in a catalog of macros. You can then use the macro in formulas and calculation scripts until the macro is removed from the catalog.

You can register a custom-defined macro in the following ways:

- As local, in which the macro is available only in the Essbase application in which it was created
- As global, in which the macro is available to all Essbase applications running on the Essbase Server where the macro was created

➤ To create a custom-defined macro, use a tool:

Tool	Topic	Location
Administration Services	Creating Custom-Defined Macros	*Oracle Essbase Administration Services Online Help*
MaxL	**create macro**	*Oracle Essbase Technical Reference*

The following MaxL statement creates a local macro named @COUNTRANGE for use in the Sample application:

```
create macro Sample.'@COUNTRANGE'(Any) AS
'@COUNT(SKIPMISSING, @RANGE(@@S))'
spec '@COUNTRANGE(MemberRange)'
comment 'counts all non-missing values';
```

The following MaxL statement creates a global macro named @COUNTRANGE:

```
create macro'@COUNTRANGE'(Any) AS
'@COUNT(SKIPMISSING, @RANGE(@@S))'
```

```
spec '@COUNTRANGE(MemberRange)'
comment 'counts all non-missing values';
```

Using Custom-Defined Macros

You can use custom-defined macros like native calculation commands in calculation scripts or formulas.

➤ To use a custom-defined macro:

1 **Create or open an existing calculation script or formula.**

- If it was registered locally, you must use a calculation script or formula within the application in which the macro was created.
- If it was registered globally, you can use any calculation script or formula within any application on the Essbase Server.

2 **Add the custom-defined macro to the calculation script or formula.**

For example, to use the @COUNTRANGE custom-defined macro shown earlier in this chapter, create the following calculation script:

```
CountMbr = @COUNTRANGE(Sales, Jan:Dec);
```

Use this calculation script with the Sample.Basic database, or replace “Sales, Jan:Dec” with a range of members in a test database.

3 **Save the calculation script or formula, and then run it as usual.**

Viewing Custom-Defined Macros

View a custom-defined macro to determine whether it has been successfully created, or whether it is local or global.

➤ To view a custom-defined macro, use a tool:

Tool	Topic	Location
Administration Services	Viewing Custom-Defined Macros	*Oracle Essbase Administration Services Online Help*
MaxL	**display macro**	*Oracle Essbase Technical Reference*

Updating Custom-Defined Macros

➤ To update a custom-defined macro:

1 **Determine whether the macro is registered locally or globally.**

See “Viewing Custom-Defined Macros” on page 489.

2 **To update the macro definition; use a tool:**

Tool	Topic	Location
Administration Services	Editing Custom-Defined Macros	*Oracle Essbase Administration Services Online Help*
MaxL	**create macro** or **replace macro**	*Oracle Essbase Technical Reference*

The following MaxL statement changes the local macro @COUNTRANGE, which is used only in the Sample application:

```
create or replace macro Sample.'@COUNTRANGE'(Any)
as '@COUNT(SKIPMISSING, @RANGE(@@S))';
```

The following MaxL statement changes the global macro @COUNTRANGE:

```
create or replace macro '@COUNTRANGE'(Any)
as '@COUNT(SKIPMISSING, @RANGE(@@S))';
```

Copying Custom-Defined Macros

You can copy custom-defined macros to any Essbase Server and application to which you have appropriate access.

➤ To copy a custom-defined macro, use a tool:

Tool	Topic	Location
Administration Services	Copying Custom-Defined Macros	*Oracle Essbase Administration Services Online Help*
MaxL	**create macro**	*Oracle Essbase Technical Reference*

Removing Custom-Defined Macros

The procedure for removing global custom-defined macros, which is more complex than that for removing local custom-defined macros, should be performed only by DBAs.

➤ To remove a custom-defined macro:

1 **Determine whether the macro is registered locally or globally.**

See "Viewing Custom-Defined Macros" on page 489.

2 **Verify that no calculation scripts or formulas are using the custom-defined macro.**

3 **To remove the macro from the catalog of macros; use a tool:**

Tool	Topic	Location
Administration Services	Deleting Custom-Defined Macros	*Oracle Essbase Administration Services Online Help*

Tool	Topic	Location
MaxL	**drop macro**	*Oracle Essbase Technical Reference*

4 **Restart all applications associated with the macro.**

See "Refreshing the Catalog of Custom-Defined Macros" on page 491.

The following MaxL statement removes the local macro @COUNTRANGE, which is used only in the Sample application:

```
drop macro Sample.'@COUNTRANGE';
```

The following MaxL statement removes the global macro @COUNTRANGE:

```
drop macro '@COUNTRANGE';
```

Refreshing the Catalog of Custom-Defined Macros

Refresh the catalog of custom-defined macros when you add, update, or remove macros.

➤ To refresh the catalog of custom-defined macros for all applications on a server, restart the server.

➤ To refresh the catalog of custom-defined macros for one application, use the **refresh custom definitions** MaxL statement.

The following MaxL statement refreshes the catalog of custom-defined macros for the Sample application:

```
refresh custom definition on application sample;
```

31 Developing Custom-Defined Calculation Functions

In This Chapter

The information in this chapter applies only to block storage databases and is not relevant to aggregate storage databases. Also see Chapter 57, "Comparison of Aggregate and Block Storage."

Process for Creating Custom-Defined Functions

To enhance the calculation functions provided with Essbase, you can create custom-defined functions for use in calculation scripts. Essbase does not provide sample custom-defined functions.

Custom-defined calculation functions are written in Java. Essbase does not provide tools for creating Java classes and archives; you must have a supported version of the JDK. See the *Oracle Hyperion Enterprise Performance Management System Installation Start Here*.

For examples of custom-defined functions, see the *Oracle Essbase Technical Reference*.

➤ To create a custom-defined function:

1 **Review the requirements for custom-defined functions.**

See "Custom-Defined Function Requirements" on page 494.

2 **Write a public Java class that contains at least one public, static method to be used as a custom-defined function.**

See “Creating and Compiling a Java Class” on page 495.

3 **Install the Java class.**

See “Installing Java Classes on Essbase Server” on page 496.

4 **Register the custom-defined function as a local or global function.**

See “Registering Custom-Defined Functions” on page 497.

Custom-Defined Function Requirements

You can create more than one method in a class for use as a custom-defined function. Typically, Oracle recommends that you create the methods that you plan to use across all applications on an Essbase Server as custom-defined functions in a single class. If, however, you plan to add custom-defined functions that will be used in selective applications on the Essbase Server, create these custom-defined functions in a separate class and add them to Essbase Server in a separate `.jar` file.

When creating multiple Java classes that contain methods for use as custom-defined functions, verify that each class name is unique. Duplicate class names cause methods in the duplicate class not to be recognized, and you cannot register those methods as custom-defined functions.

Using test programs in Java, test the Java classes and methods. When you are satisfied with the output of the methods, install them on Essbase Server and register them in a single test application. Do not register functions globally for testing; doing so makes updating them more difficult if you encounter problems.

Methods in custom-defined functions can have any combination of the following supported data types as input parameters:

- boolean
- byte
- char
- com.hyperion.essbase.calculator.CalcBoolean
- float, double
- java.lang.String
- short, int, long
- arrays of any of these types

CalcBoolean is an Essbase-specific data type that can include three values—TRUE, FALSE, and #MISSING. For information about the other listed data types, see the JDK documentation.

The method return data type can be void or any of the preceding data types. Returned data types are converted to Essbase-specific data types. Strings are mapped to a string type. Boolean values are mapped to the CalcBoolean data type. All other values are mapped to a double type.

Note:

Essbase does not support double variables returned with infinite or Not-a-Number values. If these values are returned from a Java program, they may not be recorded or displayed correctly in Essbase. Double variables should be checked for infinite or Not-a-Number values and set to finite values before being returned to Essbase. See the entry for the class, Double, in the JDK documentation.

For creating, deleting, and managing custom-defined functions, Essbase requires these security permissions:

- Local, application-wide, custom-defined functions: Application Manager or higher
- Global, server-wide, custom-defined functions: Administrator

When you register a custom-defined function in Essbase, you give the function a name, which is used in calculation scripts and formulas and is distinct from the Java class and method name used by the function.

Follow these requirements for naming custom-defined functions:

- Start the name with the @ symbol. The rest of a function name can contain letters, numbers, and the following symbols: @, #, $, and _. Function names cannot contain spaces.

  For example: @MYFUNCTION
- Start the names of custom-defined functions that are called only by custom-defined macros with "@_", to distinguish them from general-use functions and macros.

  For example: @_MYFUNCTION
- Custom-defined functions must have unique names. Function names must be different from each other, from the names of custom-defined macros, and from the names of existing calculation functions.
- If an Essbase application contains a local function that has the same name as a global function, the local function is used for calculation.

Creating and Compiling a Java Class

To create and compile a Java class, use a text editor and the JDK `javac` tool.

➤ To create a Java class for a custom-defined function:

1 **In a text editor, create a Java class.**

For example:

```
public class CalcFunc {
  public static double sum (double[] data) {
    int i, n = data.length;
    double sum = 0.0d;
    for (i=0; i<n; i++) {
      double d = data [i];
      sum = sum + d;
    }
    return sum;
  }
}
```

2 **Save the file with a `.java` extension.**

For example:

`CalcFunc.java`

3 **Navigate to the directory where the `.java` file resides; at a command prompt, enter this command:**

```
javac java_filename
```

For example:

```
javac CalcFunc.java
```

4 **Resolve any compiling errors until the compiler creates a new file with a `.class` extension.**

For example:

`CalcFunc.class`

Installing Java Classes on Essbase Server

Java classes must be compiled in a JAR file, using the JDK `jar` tool.

➤ To create a `.jar` file and install it on an Essbase Server:

1 **Navigate to the directory where the `.class` file resides; at a command prompt, enter this command:**

```
jar cf jar_filename class_filename
```

For example:

```
jar cf CalcFunc.jar CalcFunc.class
```

2 **On the computer running Essbase Server, copy the `.jar` file to one of the following directories (if the directory does not exist, create it):**

- For `.jar` files containing global custom-defined functions:

  `ARBORPATH/java/udf/`

- For `.jar` files to be used only with specific applications:

  `ARBORPATH/app/AppName/udf/`

  where *AppName* is the name of the application where the local custom-defined function will be used.

If the `.jar` file is subsequently placed in another location, you must modify the *CLASSPATH* variable to include the full path and filename for the `.jar` file.

3 **If the functions will be used only by specific applications, restart those applications in Essbase. Otherwise, restart Essbase Server.**

Registering Custom-Defined Functions

After you have compiled the Java classes for custom-defined functions into `.jar` files and installed the `.jar` files on Essbase Server, you must register the custom-defined functions before you can use them in calculation scripts and formulas. See "Custom-Defined Function Requirements" on page 494.

When you register a global custom-defined function, all Essbase applications on the Essbase Server can use it. Test custom-defined functions in a single application (and register them only in that application) before making them global functions.

Use the same process for updating the catalog of functions as for updating the catalog of macros. See "Refreshing the Catalog of Custom-Defined Macros" on page 491.

Caution!

Do not register global functions for testing; doing so makes changing them more difficult if you encounter problems.

➤ To register a custom-defined function, use a tool:

Tool	Topic	Location
Administration Services	Creating Custom-Defined Functions	*Oracle Essbase Administration Services Online Help*
MaxL	**create function**	*Oracle Essbase Technical Reference*

To register a custom-defined function with local scope, include the application name as a prefix. For example, the following MaxL statement registers the custom-defined function, @JSUM, in the CalcFunc class as a local function for use within the Sample application:

```
create function Sample.'@JSUM'
as 'CalcFunc.sum'
spec '@JSUM(memberRange)'
comment 'adds list of input members';
```

To register a custom-defined function with global scope, do not include the application name as a prefix. For example, the following MaxL statement registers the custom-defined function, @JSUM, in the CalcFunc class as a global function for use in any application on Essbase Server:

```
create function '@JSUM'
as 'CalcFunc.sum'
spec '@JSUM(memberRange)'
comment 'adds list of input members';
```

Note:

Specifying input parameters for the Java method is optional. If you do not specify input parameters, Essbase reads them from the method definition in the Java code. If, however, you are registering multiple custom-defined functions with the same method name but with different parameter sets, you must register each version of the function separately, specifying the parameters for each version of the function.

Using Registered Custom-Defined Functions

You can use registered custom-defined functions like native Essbase calculation commands.

➤ To use a registered custom-defined function:

1 **Create or open an existing calculation script or formula.**

- If the custom-defined function was registered locally—within a specific application—you must use a calculation script or formula within that application.
- If the custom-defined function was registered globally, you can use any calculation script or formula on Essbase Server.

2 **Add the custom-defined function to the calculation script or formula.**

For example, to use JSUM, use this calculation script:

```
"New York" = @JSUM(@LIST(2.3, 4.5, 6.6, 1000.34));
```

Use this calculation script with the Sample.Basic sample database, or replace "New York" with the name of a member in a test database.

3 **Save the calculation script or formula, and then run it as usual.**

Updating Custom-Defined Functions

The procedure for updating custom-defined functions depends on these conditions:

- Whether the function is registered locally or globally
- Whether the signature of the custom-defined function—class name, method name, or input parameters— has been changed in the Java code for the custom-defined function

Typically, to update a custom-defined function, you must replace the `.jar` file that contains the code for the function, and then re-register the function. If, however, the signature of the custom-defined function has not changed, and the function has only one set of input parameters (it is not an overloaded method), you can replace the `.jar` file that contains the function.

Note:

Only DBAs should update global custom-defined functions.

➤ To update a custom-defined function:

1 **Determine whether the function is local or global.**

See "Viewing Custom-Defined Functions" on page 499.

2 **Make the changes to the Java class for the custom-defined function and use Java test programs to test its output.**

3 **Compile the Java classes and archive them in a new `.jar` file, using the same name as the previous `.jar` file.**

Include any other classes and methods for custom-defined functions that were included in the previous `.jar` file.

4 **Perform an action, depending on whether you are updating a local or global custom-defined function:**

a. Local: Shut down any Essbase applications that use the functions in the `.jar` file.

b. Global: Shut down all Essbase applications

If you are unsure which Essbase applications use which functions in the `.jar` file, shut down all Essbase applications.

5 **Copy the new `.jar` file to Essbase Server, replacing the existing `.jar` file with the same name.**

6 **If the signature of the custom-defined function has not changed, skip to step 8.**

7 **To replace the custom-defined function, use a tool:**

- Administration Services: See "Editing Custom-Defined Functions" in the *Oracle Essbase Administration Services Online Help*.
- MaxL: Use the **create or replace function** statement. For example:
  - Local:

    ```
    create or replace function sample.'@JSUM'
    as 'CalcFunc.sum';
    ```
  - Global:

    ```
    create or replace function '@JSUM'
    as 'CalcFunc.sum';
    ```

8 **Restart the applications that you shut down, which updates the catalog.**

Viewing Custom-Defined Functions

You can view custom-defined functions to determine whether a function has been registered successfully and whether it is registered locally or globally. Custom-defined functions are not displayed until they have been created and registered.

➤ To view a custom-defined function, use a tool:

Tool	Topic	Location
Administration Services	Viewing Custom-Defined Functions	*Oracle Essbase Administration Services Online Help*
MaxL	**display function**	*Oracle Essbase Technical Reference*

For example, use the following MaxL statement to view the custom-defined functions in the Sample application and any registered global functions:

```
display function Sample;
```

The **display function** statement lists global functions without an application name to indicate that they are global. If the application contains a function with the same name as a global function, only the local function is listed.

Removing Custom-Defined Functions

The following permissions are required to remove a custom-defined function:

- Local: Application Manager permission for the application, or any wider permission
- Global: Administrator permission

Before removing custom-defined functions, you should verify that no calculation scripts or formulas are using them. Global custom-defined functions can be used in calculation scripts and formulas across Essbase Server, so you must verify that no calculation scripts or formulas on Essbase Server are using a global custom-defined function before removing it.

Caution!

Remove global custom-defined functions only when users are not accessing Essbase databases and calculation routines are not being performed.

➤ To remove a custom-defined function:

1 **Determine whether the function is local or global.**

See "Viewing Custom-Defined Functions" on page 499.

2 **Perform an action, depending on whether you are removing a local or global custom-defined function:**

a. Local: Shut down any Essbase applications that use the functions in the `.jar` file.

b. Global: Shut down all Essbase applications

3 **To remove the custom-defined function, use a tool:**

- Administration Services: See "Deleting Custom-Defined Functions"

  in the *Oracle Essbase Administration Services Online Help*
- MaxL: Use the **drop function** MaxL statement. For example:
- Local:

  ```
  drop function Sample.'@JSUM';
  ```
- Global:

  ```
  drop function '@JSUM';
  ```

4 **Restart the applications that you shut down, which updates the catalog.**

Copying Custom-Defined Functions

You can copy custom-defined functions to any Essbase Server and application to which you have appropriate access.

To copy a custom-defined function, use a tool:

Tool	Topic	Location
Administration Services	Copying Custom-Defined Functions	*Oracle Essbase Administration Services Online Help*
MaxL	**create function**	*Oracle Essbase Technical Reference*

Performance Considerations for Custom-Defined Functions

Because custom-defined functions are implemented as an extension of the Essbase calculator framework, you can expect custom-defined functions to operate less efficiently than functions native to the Essbase calculator framework.

To optimize performance, limit the use of custom-defined functions to calculations that you cannot perform with native Essbase calculation commands, particularly in applications where calculation speed is critical.

Memory Considerations for Custom-Defined Functions

Use of the JVM API and Java API for XML Parsing has an initial effect on the memory required to run Essbase. The memory required to run these additional components is documented in the memory requirements for Essbase. For information about memory requirements, see the *Oracle Hyperion Enterprise Performance Management System Installation Start Here*.

Beyond these startup memory requirements, the Java programs you develop for custom-defined functions sometimes require additional memory. When started, the JVM for Win32 operating systems immediately allocates 2 MB of memory for programs. This allocation is increased according to the requirements of the programs that are then run by the JVM. The default upper limit of memory allocation for the JVM on Win32 operating systems is 64 MB. If the execution of a Java program exceeds the default upper limit of memory allocation for the JVM, the JVM generates an error. For more information about JVM memory management and memory allocation details for other operating systems, see JDK documentation.

Considering the default memory requirements of the JVM and the limitations of the hardware on which you run servers, carefully monitor your use of memory. In particular, developers of custom-defined functions should be careful not to exceed memory limits of the JVM when creating large artifacts within custom-defined functions.

Part V

Retrieving Data

In Retrieving Data:

32 Understanding Report Script Basics

In This Chapter

Working With a Simple Report Script

When you combine report commands that include page, row, and column dimension declarations with selected members, you have all the elements of a simple report script.

The following step-by-step example of the report script process includes a sample script. that specifies these elements, dimensions, and member selection commands. It includes comments, which document the behavior of the script, and the **!** output command. The script is based on the Sample.Basic database, which is supplied with the Essbase Server installation.

1. Create a report script.

   See "Creating Scripts" in the *Oracle Essbase Administration Services Online Help*.

2. Type the following information in the report script.

```
// This is a simple report script example
// Define the dimensions to list on the current page, as below
<PAGE (Market, Measures)

// Define the dimensions to list across the page, as below
<COLUMN (Year, Scenario)

// Define the dimensions to list down the page, as below
<ROW (Product)

// Select the members to include in the report
Sales
```

```
<ICHILDREN Market
Qtr1 Qtr2
Actual Budget Variance
<ICHILDREN Product

// Finish with a bang
!
```

3. Save the report script.

   See "Saving Scripts" in the *Oracle Essbase Administration Services Online Help*.

4. Execute the report script.

   See "Executing Report Scripts" in the *Oracle Essbase Administration Services Online Help*.

When you execute the example script against the Sample.Basic database, it produces the following report:

```
                                       East Sales

                             Qtr1                       Qtr2
                Actual    Budget  Variance    Actual    Budget  Variance
              ======== ======== ======== ======== ======== ========
100              9,211     6,500     2,711    10,069     6,900     3,169
200              6,542     3,700     2,842     6,697     3,700     2,997
300              6,483     4,500     1,983     6,956     5,200     1,756
400              4,725     2,800     1,925     4,956     3,200     1,756
  Product       26,961    17,500     9,461    28,678    19,000     9,678

                                       West Sales

                             Qtr1                       Qtr2
                Actual    Budget  Variance    Actual    Budget  Variance
              ======== ======== ======== ======== ======== ========
100              7,660     5,900     1,760     7,942     6,500     1,442
200              8,278     6,100     2,178     8,524     6,200     2,324
300              8,599     6,800     1,799     9,583     7,600     1,983
400              8,403     5,200     3,203     8,888     6,300     2,588
  Product       32,940    24,000     8,940    34,937    26,600     8,337

                                      South Sales

                             Qtr1                       Qtr2
                Actual    Budget  Variance    Actual    Budget  Variance
              ======== ======== ======== ======== ======== ========
100              5,940     4,100     1,840     6,294     4,900     1,394
200              5,354     3,400     1,954     5,535     4,000     1,535
300              4,639     4,000       639     4,570     3,800       770
400           #Missing #Missing #Missing #Missing #Missing #Missing
  Product       15,933    11,500     4,433    16,399    12,700     3,699

                                     Central Sales

                             Qtr1                       Qtr2
                Actual    Budget  Variance    Actual    Budget  Variance
              ======== ======== ======== ======== ======== ========
100              9,246     6,500     2,746     9,974     7,300     2,674
200              7,269     6,800       469     7,440     7,000       440
```

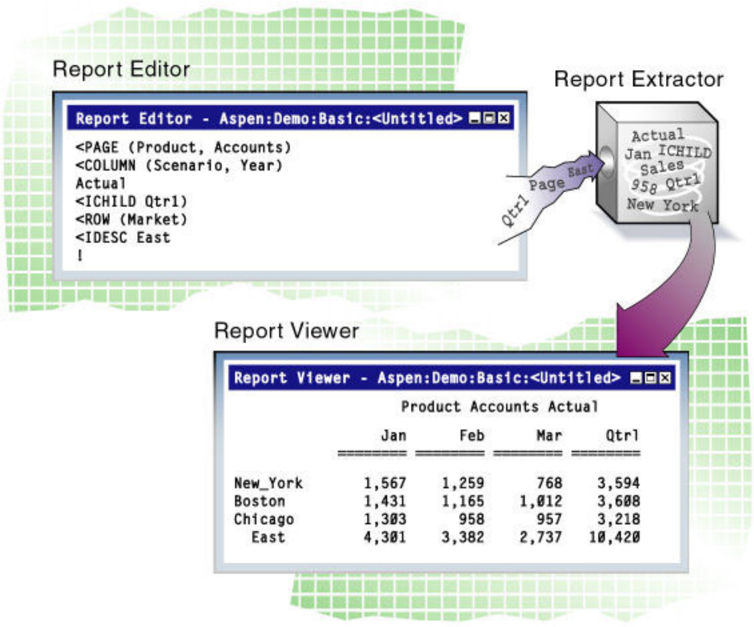
```
300                 10,405    6,200    4,205   10,784    6,800    3,984
400                 10,664    5,200    5,464   11,201    5,800    5,401
  Product           37,584   24,700   12,884   39,399   26,900   12,499

                                 Market Sales

                             Qtr1                        Qtr2
                  Actual   Budget  Variance   Actual   Budget  Variance
                ======== ======== ======== ======== ======== ========
100                 32,057   23,000    9,057   34,279   25,600    8,679
200                 27,443   20,000    7,443   28,196   20,900    7,296
300                 30,126   21,500    8,626   31,893   23,400    8,493
400                 23,792   13,200   10,592   25,045   15,300    9,745
  Product          113,418   77,700   35,718  119,413   85,200    34,21
```

Understanding How Report Writer Works

Report Writer comprises three main components:

- Report Script Editor is a text editor that you use to write report scripts. Report commands define formatted reports, export data subsets from a database, and produce free-form reports. Execute the saved script to generate a report. Saved report scripts have the file extension `.rep`.
- Report Extractor retrieves data information from the Essbase database when you run a report script.
- Report Viewer displays the complete report. Saved reports have the file extension `.rpt`.

Figure 141 Report Writer Components



```
<PAGE (Product, Accounts)
<COLUMN (Scenario, Year)
Actual
<ICHILD Qtr1)
<ROW (Market)
<IDESC East
!
```

```
                 Product Accounts Actual

                Jan      Feb      Mar     Qtr1
           ======== ======== ======== ========

New_York      1,567    1,259      768    3,594
Boston        1,431    1,165    1,012    3,608
Chicago       1,303      958      957    3,218
  East        4,301    3,382    2,737   10,420
```

Report Extractor

➤ The Report Extractor processes the report script and retrieves data, performing the following actions:

1. **Composes the member list, based on all possible member combinations. For example, `<IDESCENDANTS East` retrieves member East and all of its descendants.**
2. **Applies member restrictions. For example, `<LINK` refines the member selection.**
3. **Orders the member output. For example, `<SORT` determines the order in which members are sorted.**
4. **Extracts data from the following areas:**
   - Local regions
   - Partitioned regions
   - Dynamically calculated data
5. **Restricts data. For example, the following command suppresses the display of all rows that contain only missing values:**

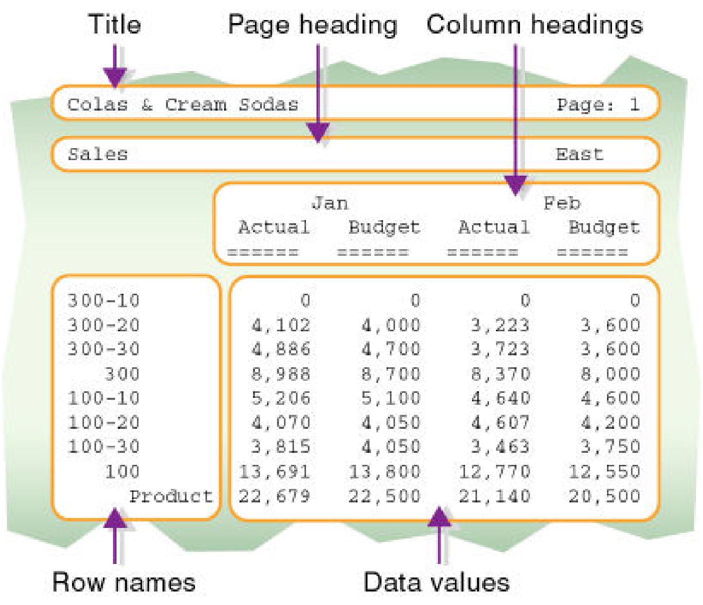
   ```
   {SUPMISSINGROWS}
   ```
6. **Sorts data. For example, `<TOP` returns rows with the greatest values of a specified data column.**
7. **Formats output. For example, `{SKIP}` skips one or more lines in the final output report.**

The order in which Report Extractor retrieves data affects the execution of complex extraction and formatting commands. For example, because the Report Extractor restricts data (step 5) before sorting data (step 6), if you place conditional retrieval commands in the wrong order, report output results can be unexpected. Be aware of the data retrieval process when designing report scripts.

Parts of a Report

Understanding the parts of a report is essential as you plan and design your own reports.

Figure 142   Elements of a Typical Report



```
Colas & Cream Sodas                               Page: 1
Sales                                             East
                      Jan                 Feb
            Actual    Budget    Actual    Budget
            ======    ======    ======    ======
300-10           0         0         0         0
300-20       4,102     4,000     3,223     3,600
300-30       4,886     4,700     3,723     3,600
   300       8,988     8,700     8,370     8,000
100-10       5,206     5,100     4,640     4,600
100-20       4,070     4,050     4,607     4,200
100-30       3,815     4,050     3,463     3,750
   100      13,691    13,800    12,770    12,550
     Product 22,679   22,500    21,140    20,500
```

A typical report is composed of the following parts:

- Page Headings list dimensions represented on the current page. All data values on the page have the dimensions in the page heading as a common property.

  ```
  <PAGE (Market, Measures)
  ```

- *Column Headings* list members across a page. You can define columns that report on data from more than one dimension, which results in *nested column headings*.

  ```
  <COLUMN (Year, Scenario)
  ```

- Row Headings list members down a page. You can define a member list that includes rows from more than one level within a dimension or from more than one dimension. The rows are indented below the dimension name.

  ```
  <ROW (Product)
  ```

- Titles contain user-defined text, date and time stamp, the user name of the person running the report, page numbers, the name of the source database, or any other descriptive information. Titles are user-generated and optional. Page, column, and row headings are automatically generated, because they are necessary to clearly describe the data on the report page.

  ```
  { STARTHEADING TEXT 1 "Prepared by:" 14 "*USERNAME" C "The
  Electronics Club" 65 "*PAGESTRING" TEXT 65 "*DATE" SKIP ENDHEADING }
  ```

- Data values are the values contained in the database cells; they are the lookup results of member combinations or the results of calculations when the report is run through the Report Extractor. Each data value is the combination of the members in the page heading, column heading, and row name.

  All data values in a row share the properties of the row names of that row. A report can have zero or more row name dimensions, each of which produces column of row names, with the innermost row name column cycling the fastest.

Parts of a Report Script

A report script comprises a series of Report Writer commands, terminated by the bang (!) report output command.

You can enter one or more report scripts in a report script file, which is a text file that you create with Report Script Editor or any text editor.

To build a report script, enter or select commands that define the layout, member selection, and format in Report Script Editor. The different elements of a script are color-coded to aid readability. You can enable syntax autocompletion to help build scripts quickly.

The commands in Report Writer perform two functions, data extraction and formatting:

- Extraction commands deal with the selection, orientation, grouping, and ordering of raw data extracted from the database. These commands begin with less-than signs (<).
- Formatting commands allow customization of the report format and appearance, the creation of new columns, and calculation of columns and rows. These commands are generally contained within braces ({ }), although some begin with less-than signs (<).
- The *bang* character (!) terminates a series of commands and requests information from the database. You can place one or more report scripts, each terminated by its own ! command, in the same report file.

For information about report commands, see the *Oracle Essbase Technical Reference*.

Planning Reports

Report design is important to presenting information. Include the proper elements and arrange information in an attractive, easy-to-read layout.

➤ To plan a report:

1. **Consider the reporting needs and the time required to generate the report.**
2. **Roughly sketch the report. Include the following items:**
   - Layout
   - Number of columns
   - Members
   - Titles, if applicable
   - Format of data values
3. **Review the sketch; it is often apparent at this stage if additional data or formatting is needed.**
4. **Determine ways to optimize the report runtime.**

   See Chapter 56, “Optimizing Reports and Other Types of Retrieval” for a comprehensive discussion of how to optimize a report script.

Note:

As you plan the report, minimize use of numeric row names; instead, to avoid ambiguity, give rows descriptive names.

Considering Security and Multiple-User Issues

You must use Administration Services to use Report Script Editor to create or modify a report script. You can also use any text editor to create script files. If you use Report Script Editor, it enables you to create and modify report scripts stored on your desktop machine, as well as the Essbase Server. To modify report scripts stored on the server, you must have Application Manager or Database Manager access.

Essbase supports concurrent, multiple-user database access. As in most multiple-user environments, Essbase protects critical data with a security system. Users can read or update data only with the correct permissions.

When you execute a report script, Essbase security verifies that you have read or higher access level to all data members specified in the report. In a filtering process identical to that for retrieving members into a spreadsheet, Essbase filters any member from the output for which you have insufficient permissions.

To users who are only reporting data, locks placed by other users are transparent. Even if a user has locked and is updating part of the data required by the report, the lock does not interfere with the report in any way. The data in the report reflects the data in the database at the time you run the report. Running the same report later reflects any changes made after the last report ran.

See Chapter 38, "User Management and Security" for a comprehensive discussion of the Essbase security system.

Reviewing the Process for Creating Report Scripts

This section describes the process for creating a report script.

1. Create the report script.

   See "Creating Report Scripts" on page 512.
2. Check the report script syntax.

   See "Checking Script Syntax" in the *Oracle Essbase Administration Services Online Help*.
3. Save the report script.

   See "Saving Report Scripts" on page 512.
4. Run the report script.

   See "Executing Report Scripts" on page 512.
5. If desired, save the report.

   See "Saving Reports" in the *Oracle Essbase Administration Services Online Help*.

Creating Report Scripts

You can report on the data in a database using the following methods:

- Report Script Editor. Use Report Script Editor to create large-scale reports comprising many pages of multidimensional data. Reports of this scale often can exceed the capabilities of even the most robust spreadsheet. Report Writer commands let you define formatted reports, export data subsets from an Essbase database, and produce free-form reports. See "Creating Scripts" in the *Oracle Essbase Administration Services Online Help*.
- A text editor.
- Through a spreadsheet. Use report commands in a spreadsheet in free-form mode or template-retrieval mode. See the *Oracle Essbase Spreadsheet Add-in User's Guide*.
- Essbase APIs. See "Generating Reports Using the C, Visual Basic, and Grid APIs" on page 550 and see the *Oracle Essbase API Reference*.
- Third-party reporting tools.

For information about creating and editing report scripts in Administration Services, see "About Report Script Editor" in the *Oracle Essbase Administration Services Online Help*.

Saving Report Scripts

You can save a report script in the following locations:

- As a file on a client machine or network.
- As an artifact on Essbase Server. To allow other users access to the report script, save it on Essbase Server. You can associate the script artifact with the following artifacts:
  - An application and all the databases within the application, which lets you run the script against any database in the application.
  - A database, which lets you run the script against the specified database.

Report scripts have a `.rep` extension by default. If you run a report script from Administration Services, the script must have a `.rep` extension.

➤ To save a report script using Report Script Editor, see "Saving Report Scripts" in the *Oracle Essbase Administration Services Online Help*.

Executing Report Scripts

When you execute a report script using Administration Services, you can send the results to the Report Viewer window, to a printer, and/or to a file. From the Report Viewer window, you can print, save, and copy the report.

Using Administration Services, you can execute a report in the background so that you can continue working as the report processes. You can then check the status of the background process to see when the report has completed.

See “Executing Report Scripts” in the *Oracle Essbase Administration Services Online Help*.

Copying Report Scripts

You can copy report scripts to applications and databases on any Essbase Server, according to your permissions. You can also copy scripts across servers as part of application migration.

➤ To copy a report script, use a tool:

Tool	Topic	Location
Administration Services	Copying Scripts	*Oracle Essbase Administration Services Online Help*
MaxL	**alter object**	*Oracle Essbase Technical Reference*
ESSCMD	COPYOBJECT	*Oracle Essbase Technical Reference*

Developing Free-Form Reports

Free-form reports are often easier to create than structured reports. The free-form reporting style is ideal for ad hoc reporting in the Report Script Editor window.

A free-form report does not include PAGE, COLUMN, or ROW commands and instead gathers this information from a series of internal rules that are applied to the report script by the Report Extractor when you run the report.

The following example script and report illustrate free-form reporting:

```
Sales Colas
Jan Feb Mar
Actual Budget
Illinois
Ohio
Wisconsin
Missouri
Iowa
Colorado
{UCHARACTERS}
Central
     !
```

This example produces the following report:

```
                  Sales 100

           Jan                Feb                Mar
      Actual   Budget   Actual  Budget   Actual  Budget
     =======  =======   ======  ======   ======  ======
Illinois  829      700      898     700      932     700
Ohio      430      300      397     300      380     300
Wisconsin 490      300      518     400      535     400
Missouri  472      300      470     300      462     300
```

```
Iowa          161       0     162       0     162       0
Colorado      643     500     665     500     640     500
========      ===     ===     ===     ===     ===     ===
 Central    3,025   2,100   3,110   2,200   3,111   2,200
```

You can use formatting commands to add specific formats to a free-form report. When PAGE, COLUMN, and ROW commands are omitted, Essbase formats free-form reports according to the following rules:

1. The Report Extractor finds the last member or members of a single dimension defined in the report specification (before the report output operator !). This dimension becomes the ROW dimension for the report. All remaining selections become PAGE or COLUMN dimensions, as defined by rules 2 and 3.
2. The Report Extractor searches for any single-member selections. If a single member is found that does not satisfy rule 1, that dimension becomes a PAGE dimension.
3. The Report Extractor searches for all remaining dimension members that do not satisfy rules 1 or 2. If any remaining members are found, those dimensions become COLUMN dimensions. COLUMN dimensions are nested in the order of selection in the free-form script.
4. The Report Extractor searches the database outline for any dimensions not specified in the report specification. If unspecified dimensions are found, they become PAGE dimensions (the default for single-member selections, as defined in rule 2).
5. A subsequent selection of one or more consecutive members from a given dimension overrides any previous selection for that dimension.

For example, the following report recognizes California, Oregon, Washington, Utah, Nevada, and West as members of Market.

```
Sales
Jan Feb Mar
Actual Budget
Apr May Jun
California
Oregon
Washington
Utah
Nevada
{UCHARACTERS}
West
    !
```

Report Extractor applies free-form formatting rules to this report:

1. Because California, Oregon, Washington, Utah, Nevada, and West are listed last, the Report Extractor treats them as if ROW (Market) had been specified (rule 1).
2. Sales is a single-member selection from dimension Measures. The Report Extractor treats this member as if PAGE (Measures) had been specified (rule 2).
3. After searching the remaining members, the Report Extractor finds members of dimensions Year and Scenario, which it treats as COLUMN (Year, Scenario) (rule 3).

4. The Report Extractor searches the database outline and finds that dimension Product is not specified in the report specification. Because Product is a single-member selection, the Report Extractor treats this member as if PAGE (Product) had been specified (rule 4).
5. Finally, the Report Extractor finds that Apr May Jun is from the same dimension as Jan Feb Mar and is displayed on a subsequent line of the script. The Report Extractor discards the first specification (Jan Feb Mar) and uses the second (Apr May Jun).

As a result, the report example produces the following report:

```
                     Product Sales

                Actual                     Budget
             Apr    May     Jun     Apr     May     Jun
            ======= ====== ======  ======  ======  ======
California  3,814   4,031   4,319   3,000   3,400   3,700
Oregon      1,736   1,688   1,675   1,100   1,000   1,100
Washington  1,868   1,908   1,924   1,500   1,600   1,700
Utah        1,449   1,416   1,445     900     800     800
Nevada      2,442   2,541   2,681   1,900   2,000   2,100
======      =====   =====   =====   =====   =====   =====
  West     11,309  11,584  12,044   8,400   8,800   9,400
```

Note:

You cannot use substitution variables in free-form mode.

33 Developing Report Scripts

In This Chapter

Also see:

- Chapter 32, "Understanding Report Script Basics"
- *Oracle Essbase Technical Reference*, for report command syntax and an extensive set of report script examples

Introduction

When you understand the basics of creating report scripts, you can create more complex reports. You create a report using *extraction commands*, which specify member combinations for pages, columns, and rows. You use *formatting commands* to determine the visual design of the report and to control the display of the data values. Formatted data values are displayed in the report when you run the script, based on the combined extraction and report commands.

Understanding Extraction and Formatting Commands

Extraction commands perform the following actions:

- Determine the selection, orientation, grouping, and ordering of raw data records extracted from the database. Extraction commands are based on either dimension or member names, or keywords. Their names begin with the greater-than symbol (>).

- Apply to the report from the line on which they occur until the end of the report. If another extraction command occurs on a subsequent line of the report, it overrides the previous command.

Formatting commands perform the following actions:

- Enable you to customize the format and appearance of a report and create report-time calculations. Formatting commands are generally enclosed in braces ({ }), although several formatting commands begin with the less-than (<) character.
- Are either applied globally within the report script or are specific to a member.

Understanding Report Script Syntax

To build a report, you enter commands that define the layout, member selection, and format you want in Report Script Editor. The different elements of a script are color-coded to aid readability. You can enable autocompletion to help build scripts interactively as you type. See "About Report Script Editor" in the *Oracle Essbase Administration Services Online Help*.

When you write a report script, follow these guidelines:

- Separate commands with at least one space, tab, or new line for readability. Report processing is unaffected by extra blank lines, spaces, or tabs.
- Enter commands in uppercase or lowercase. Commands are not case-sensitive. If the database outline is case-sensitive, the members in the report script must match the outline.
- To start report processing, enter the bang (!) report output command or one or more consecutive numeric values. You can place one or more report scripts, each terminated by its own ! command, in the same report file.
- You can group more than one format command within one set of braces. For example, these formats are synonymous:

  ```
  {UDATA SKIP}
   {UDATA} {SKIP}
  ```
- Enclose member names in quotation marks in the following cases:
  - Names beginning with an ampersand (for example, "&Product").
  - Names containing spaces (for example, "Cost of Goods Sold").
  - Names containing the word Default (for example, "Default Value").
  - Duplicate member names, which must be entered as qualified member names (for example, "[2006].[Qtr1]")
  - Names containing one or more numerals at the beginning of the name (for example, "100-Blue")
  - Names containing any of the following characters:

*	asterisks	-	dashes, hyphens, or minus signs
@	at signs	<	less-than signs

{}	braces	()	parentheses
[ ]	brackets	+	plus signs
,	commas	;	semicolons
:	colons	/	slashes

- If a formatting command is preceded by three or more underscore, equal sign, or hyphen characters, respectively, the Report Extractor assumes that the characters are extraneous underline characters and ignores them. For example, ==={SKIP 1}.
- (double slash) Use `//` to indicate a comment. Everything on the line following a comment is ignored by the Report Writer. Each line of a comment must start with a double slash, so you can include multiline comments.
- Exercise caution in abbreviating command names. Many names begin with the same letters, and the results may be unexpected unless you use a completely unambiguous abbreviation.

Designing the Page Layout

Reports are two-dimensional views of multidimensional data. You can use page layout commands to incorporate additional dimensions that are defined as nested groups of columns or rows on a page, or additional pages in the report.

The page layout is composed of headings that make up the columns and rows of a page. You define the basic layout of a report using page, row, and column data extraction commands combined with specific member selections.

Each component of page layout has a different formatting command:

```
<PAGE
<COLUMN
<ROW
```

In addition, the <ASYM and <SYM commands override the default method of interpreting the column dimension member lists, and produce either an asymmetric or symmetric report format.

See "Formatting Page, Column, and Row Headings" on page 523.

Creating Page, Column, and Row Headings

➤ To define report headings:

1 **Enter** `<PAGE(`**dimensionname**`,` **dimensionname**`)`

where *dimensionname* lists dimensions represented on the current page. All data values on the page have the dimensions in the page heading as a common property. Example:

```
<PAGE (Measures, Market)
```

2 **Press Enter.**

3 **Enter** `<COLUMN(`**dimensionname**`)`

where *dimensionname* equals the name of each dimension to display across the page. Example:

```
<COLUMN (Year)
```

Adding dimension names creates nested column headings.

4 **Press Enter.**

5 **Enter** `<ROW(`**dimensionname**`)`

where *dimensionname* equals the name of each dimension to display down the page. Example:

```
<ROW (Market)
```

6 **Press Enter.**

Note:

You can select additional members to associate with the heading commands. Using a member name as a parameter for the PAGE, COLUMN, or ROW commands causes Report Extractor to associate the member with the appropriate dimension.

The following report script is based on the Sample.Basic database:

```
<PAGE (Product, Measures)
<COLUMN (Scenario, Year)
Actual
<ICHILDREN Qtr1
<ROW (Market)
<IDESCENDANTS East
    !
```

This script produces the following report:

```
                  Product Measures Actual

                          Jan      Feb      Mar     Qtr1
                     ======== ======== ======== ========
New York                  512      601      543    1,656
Massachusetts             519      498      515    1,532
Florida                   336      361      373    1,070
Connecticut               321      309      290      920
New Hampshire              44       74       84      202
  East                  1,732    1,843    1,805    5,380
```

You can create page, column, and row headings with members of attribute dimensions. The following report script is based on the Sample.Basic database:

```
<PAGE (Measures,Caffeinated)
Profit
<COLUMN (Year,Ounces)
Apr May
"12"
<ROW (Market,"Pkg Type")
Can
<ICHILDREN East
    !
```

This script produces the following report:

```
                         Profit Caffeinated 12 Scenario
                                    Apr      May
                               ======== ========
New York         Can                276      295
Massachusetts    Can                397      434
Florida          Can                202      213
Connecticut      Can                107       98
New Hampshire    Can                 27       31
  East           Can              1,009    1,071
```

Modifying Headings

You can perform the following modifications to headings in the report:

Task	Report Command
Create a custom page heading in place of the default heading, which is displayed at the top of each page in the report or immediately following a HEADING command. Use the ENDHEADING command to specify the end of the custom heading.	STARTHEADING
Display the page heading, either the default heading or the heading as defined with the STARTHEADING and ENDHEADING commands. Use this command to re-enable the page heading display if the SUPHEADING command has been used.	HEADING
Force the immediate display of the heading without waiting for the next unsuppressed data row, when heading suppression commands are in use.	IMMHEADING
Automatically turn on the display of the column header.	COLHEADING
Display the page heading before the next data output row.	PAGEHEADING

For descriptions of the SUPPRESS commands used to suppress headings, see "Suppressing Page, Column, and Row Formatting" on page 525.

Creating Symmetric and Asymmetric Reports

Essbase reports can contain symmetric or asymmetric column groups. Essbase determines the symmetry of column groups automatically, based on the members you select.

A *symmetric report,* shown below, is characterized by repeating, identical groups of members.

```
           East                           West
   Budget        Actual         Budget       Actual
Q1   Q2   Q3    Q1   Q2   Q3       Q1   Q2   Q3   Q1   Q2   Q3
```

An *asymmetric report,* shown below, is characterized by groups of nested members that differ by at least one member in the nested group. There can be a difference in the number of members or the names of members.

```
            East                           West
   Budget         Actual                 Budget
Q1   Q2   Q3     Q1   Q2   Q3         Q1   Q2   Q3
```

By default, Essbase creates a symmetric report unless you select the same number of members for all column dimensions.

For an example of an asymmetric report, see “Sample 13, Creating Asymmetric Columns,” in the Examples of Report Scripts page of the *Oracle Essbase Technical Reference*.

The Essbase evaluation of symmetry versus asymmetry takes place before any ordering, restriction on columns, or application of the effects of calculated columns.

Overriding Default Column Groupings

You can override the default column grouping that Essbase selects for reports with the <SYM and <ASYM commands. <SYM and <ASYM affect the member selection commands that follow them in a report.

1. Use the <SYM command when the selection of column members meets the requirements of the rule for asymmetry, but you want to produce a symmetric report. The <SYM command always produces a symmetric report, creating all combinations of each column dimension.
2. Turn off the symmetric format and restore the rules for asymmetric reports with the <ASYM command.

Changing Column Headings

If you need to change only the column headings rather than the symmetry of the report, the <PYRAMIDHEADERS and <BLOCKHEADERS formatting commands are useful.

- Use the <BLOCKHEADERS formatting command to change the pyramid-style headers used in symmetric reports to block-style headers such as those used in asymmetric reports. A symmetric report uses the <PYRAMIDHEADERS mode of column layout by default.
- Use the <PYRAMIDHEADERS formatting command to change the block-style headers used in asymmetric reports to pyramid-style headers such as those used in symmetric reports. An asymmetric report uses the <BLOCKHEADERS mode of column layout.

Formatting

Formatting commands, usually enclosed in braces ({ }), define the format of data and labels in the final report and can be either *global* or *member-specific*.

- Global commands are executed when they occur in the report script file and stay in effect until the end of the report file or until another global command replaces them.

  For example, the {SUPMISSINGROWS} command suppresses all rows in the report script file that contain only missing values.
- Member-specific commands are executed as they are encountered in the report script, usually the next member in the report script, and affect only that member. A format attached to a member is executed before that member is processed.

For example, the {SKIP} command skips the specified number of rows between row dimensions in a report script. If you want additional rows to skip lines, you must use the SKIP command again.

Formatting Report Pages

You can use several formatting commands to design the look of the final report pages.

See the *Oracle Essbase Technical Reference* for examples.

Setting Page Length and Width and Centering

You can set the following page specifications in the report script:

Task	Report Command
Specify the column widths.	WIDTH
Set the left margin.	LMARGIN
Set the center of the page.	SETCENTER

Inserting Page Breaks

You can set the following types of page breaks in the report script:

Task	Report Command
Set the number of lines for each page.	PAGELENGTH
Force a page break regardless of how many lines have been generated for the current page.	NEWPAGE
Insert a page break whenever a member from the same dimension as the member in the command changes from one line in the report to the next. Use the NOPAGEONDIMENSION command to turn off this function.	PAGEONDIMENSION
Enable page breaks in a report when the number of lines on a page is greater than the current PAGELENGTH setting.	FEEDON

Formatting Page, Column, and Row Headings

Column and row formatting commands make up a special type of format setting commands.

Specifying Column Formats

Specifications for column formatting commands can precede or follow the columns to which they apply, depending on the desired format.

For example, in the following script, based on the Sample.Basic database, the first {DECIMAL} command is processed after two columns are set up, Actual and Budget. The {DECIMAL} command, however, refers to a column three, which does not yet exist. Essbase responds to this command by dynamically expanding the report to three columns. When the report specification expands to six columns, the {DECIMAL} formatting applies to columns three *and* six (and all multiples of three).

Essbase performs this pattern extension on the assumption that when another dimension is added, causing repetitions of previous column dimension groups, the formatting should repeat as well. The second {DECIMAL} formatting command is then applied to columns 1 and 4 only, as it occurs after the creation of six columns.

```
<PAGE (Measures, Market)
Texas Sales
     <COLUMN (Scenario, Year)
     Actual Budget
{DECIMAL 2 3 }
     Jan Feb Mar
{DECIMAL 1 1 4 }
<ROW (Product)
<DESCENDANTS "100"
    !
```

This script produces the following report:

```
                             Sales Texas

                  Actual                    Budget
             Jan      Feb      Mar      Jan      Feb      Mar
             ===      ===      ===      ===      ===      ===
100-10     452.0      465   467.00    560.0      580   580.00
100-20     190.0      190   193.00    230.0      230   240.00
100-30  #MISSING #MISSING #MISSING #MISSING #MISSING #MISSING
```

The following scripts demonstrate two approaches to column formatting that produce identical results. In the first script, the first two {DECIMAL} commands are positioned to format every first and third column by distributing the formats when Jan Feb displays *after* processing the {DECIMAL} command. These examples are based on the Sample.Basic database.

```
//Script One: Format Columns by Distributing the Formats

<PAGE (Measures, Market)
California Sales
     <COLUMN (Scenario, Year)
     Actual Budget Variance
{DECIMAL 1 1 }
{DECIMAL 2 3 }
     Jan Feb
     //     {DECIMAL 1 1 4 }   These lines are commented; the
     //     {DECIMAL 2 3 6 }   Report Extractor ignores them.
<ROW (Product)
<DESCENDANTS "100"
    !
```

The two {DECIMAL} commands are positioned to format the individual columns 1, 3, 4, and 6.

```
//  Script Two: Format Columns by Direct Assignment
```

```
<PAGE (Measures, Market)
California Sales
     <COLUMN (Scenario, Year)
     Actual Budget Variance
     //     {DECIMAL 1 1 }      These lines are commented; the
     //     {DECIMAL 2 3 }      Report Extractor ignores them.
     Jan Feb
{DECIMAL 1 1 4 7 }
{DECIMAL 2 3 6 9 }
<ROW (Product)
<DESCENDANTS "100"
    !
```

Both scripts produce the following report:

```
                    Sales California

             Actual          Budget            Variance
          Jan      Feb      Jan       Feb      Jan       Feb
        =====     ====     ====      ====    =====       ====
100-10  678.0      645   840.00     800.0    (162)   (155.00)
100-20  118.0      122   140.00     150.0     (22)    (28.00)
100-30  145.0      132   180.00     160.0     (35)    (28.00
```

Note:

By default, attribute calculation dimension members (for example, SUM, AVG) are displayed as columns. To display them in rows, you must include them in the ROW command.

Accommodating Long Column Names and Row Names

Member names too long to fit into columns are automatically truncated; the tilde character (~) signifies that part of the name is missing. Long member names are common when using aliases.

You can modify columns to display the entire member name:

Task	Report Command
Define the column width for all row members in the column.	NAMEWIDTH
Change where the row member column is displayed and shift the remaining columns left or right to allow long member names.	NAMESCOL

Suppressing Page, Column, and Row Formatting

You can suppress the display of page heading, columns, and rows in a report by using SUPPRESS commands.

Suppress	Report Command
The default column heading in the report.	SUPCOLHEADING
Rows that have only zero or missing values.	SUPEMPTYROWS

Suppress	Report Command
All rows that contain missing values. Use INCMISSINGROWS, INCZEROROWS, or INCEMPTYROWS to display rows that are empty or have missing data or zeros.	SUPMISSINGROWS
The page member heading whenever a heading is generated.	SUPPAGEHEADING
The page and column headings, all member names, page breaks, commas, and brackets in the final report. To turn on the display of columns of row member names, use the NAMESON command. To turn on the use of commas and brackets, use the COMMAS and BRACKETS commands.	SUPALL
The default heading (page header and column headers) or custom header, if defined, at the top of each page.	SUPHEADING
Row member names in the final report. Use the NAMESON command to include row member names in the report.	SUPNAMES
All output while continuing to process all operations, such as calculations and format settings. Use the OUTPUT command to reverse the actions of SUPOUTPUT.	SUPOUTPUT
Currency information when you use the CURRENCY command to convert data values in a report to a specified currency.	SUPCURHEADING

Repeating Row Names

To repeat the row member names on every line of the report, use the <ROWREPEAT command. Use the <NOROWREPEAT command to prevent row member names from being repeated on each line of the report if the row member name does not change on the next line. NOROWREPEAT is enabled by default.

Using Tab Delimiters

You can use tabs rather than spaces between columns in report scripts; for example, when you want to export report output into another form.

To replace spaces with tab delimiters, type {TABDELIMIT} anywhere in the report script.

Adding Totals and Subtotals

Column and row calculations let you create additional calculations that are not defined as part of the database outline. For example, you can use column and row calculations to create extra columns or rows in a report, based upon selected data members, and perform calculations on these or existing columns and rows.

For examples of report scripts that contain column and row calculations, see "Sample 14, Calculating Columns" on the "Example of Report Scripts" page of the *Oracle Essbase Technical Reference*.

Totaling Columns

The CALCULATE COLUMN command lets you create a report column, perform on-the-fly calculations, and display the calculation results in the newly created column.

The following table summarizes column calculation commands:

Task	Report Command
Create a report column, perform dynamic calculations, and display the calculation results in the newly created column. Use the OFFCOLCALCS command to temporarily disable column calculations in the report, and ONCOLCALCS to re-enable calculations.	CALCULATE COLUMN
Remove all column calculation definitions from the report.	REMOVECOLCALCS

CALCULATE COLUMN adds up to 499 ad hoc column calculations to a report. Each new calculated column is appended to the right of the existing columns in the order in which it is created and is given the next available column number. These columns calculate the sum of data across a range of columns or an arithmetic expression composed of simple mathematical operators.

The CALCULATE COLUMN command supports standard mathematical operations. For syntax and parameter descriptions, see the *Oracle Essbase Technical Reference*.

If you use the same name for more than one column, Essbase creates only the last column specified in the CALCULATE COLUMN command. Use a leading space with the second name (and two leading spaces with the third name, and so on) to create a unique column name.

Alternately, you can add descriptive text far enough to the right that it is truncated to the column width. You can, for example, use the names Q1 Actual and Q1 Budget to distinguish similar column names without affecting the appearance of the report. Column names are printed with right justification until the column header space is filled. Excess characters are then truncated to the right.

Divide lengthy column name labels into multiple lines. The maximum number of lines across which you can divide a label is equal to the number of column dimensions designated in the report specification. To break a column name, insert a tilde (~) in the name at the point where you want the break. You must also specify at least two members for each column dimension to use the maximum number of lines.

This example is based on the Sample.Basic database.

```
{CALCULATE COLUMN "Year to Date~Actual Total" = 1 : 2}
{CALCULATE COLUMN "Year to Date~Budget Total" = 3 : 4}
```

The example produces the following report:

```
                                  Sales East
              Actual    Year to Date      Budget     Year to Date
           Jan    Feb   Actual Total    Jan    Feb   Budget Total
         ===== ====== ============= ====== ====== =============
400-10     562    560         1,122    580    580         1,702
400-20     219    243           462    230    260           722
400-30     432    469           901    440    490         1,391
```

As a rule, in symmetric reports, if a calculated column name has fewer levels than the number of column dimensions, the previous member (to the left) of each of the column dimensions, above the top level supplied in the calculated column name, is attributed to the calculated column. If normal PYRAMIDHEADERS mode is in use, the centering of those higher-level column members shifts to the right to include the calculated column or columns. Column header members on the same level as calculated column names are not applied to the new calculated column or columns, and their centering does not shift.

If BLOCKHEADERS mode is in use; that is, if every member applying to a column is repeated above that column, the same rules apply, except that instead of shifting column header member centering, they are repeated in the appropriate higher levels of the calculated column name.

Asymmetric reports do not have groups of columns that share a member property. These reports still allow multiple-level column names up to the number of column levels defined, but member properties from preceding columns are not automatically shared and used for those levels that are undefined.

If there are fewer column header dimensions than the number of levels that you want, you can create multiline column labels. In this case, use TEXT, STARTHEADING, ENDHEADING, and other formatting commands to create a custom heading.

For the syntax and definitions of column calculation commands, see the *Oracle Essbase Technical Reference*.

Numbering Columns

If the number of regular (not calculated) columns varies in the report because multiple sections in the report have different numbers of columns, the column numbers used to identify the calculated columns shift accordingly, as illustrated:

- If the first section of a report has 12 columns (including row name columns), and three calculated columns are declared, column numbers 0–11 are the regular columns, and columns 12–14 are the calculated columns.
- If a second section of the report reduces the number of regular columns to six, then the regular columns are columns 0–5, and the same calculated columns are columns 6–8.
- Similarly, if the number of regular columns is increased, the numbering of the calculated columns starts at a higher number.

In the example, CC1, CC2, and CC3 represent the names of three calculated column names. The column numbering for a report with two different sections with varying numbers of regular columns:

```
internal
col # s: 0    1      2      3     4     5     6     7
            Jan    Feb    Mar   Apr   CC1   CC2   CC3
            ===    ===    ===   ===   ===   ===   ===
   Sales      1      3      5     3    22    55    26
   Expense    1      2      5     3    23    65    33

                 same report- new section
internal
col # s: 0    1      2      3     4     5
```

```
         Qtr1     YTD     CC1     CC2     CC3
          ===     ===     ===     ===     ===
Sales       2       9      22      57      36
Expense     4       8      56      45      33
```

If you do not want the calculated columns in the second section, or if you need a different set of column calculation, use the command REMOVECOLCALCS to clear the old ones out. You can then define new column calculations.

This example assumes that all three column calculations had no references to regular columns other than columns 1 and 2. If CC3's calculation were = 1 + 3 + 6, when the second section of the report starts, an error occurs stating that the column calculation referred to a nonexistent column (6).

Totaling Rows

Row calculations create summary rows in a report. You can use *summary rows* to calculate the sum of data across a range of rows or to calculate an arithmetic expression composed of simple mathematical operators.

The following table summarizes row calculation commands:

Task	Report Command
Create a row and associate it with a row name or label. This process is similar to declaring a variable. You can also perform simple calculations with CALCULATE ROW. For more complex calculations, use SETROWOP. See also OFFROWCALCS and ONROWCALCS.	CALCULATE ROW
Temporarily disable row calculations in the report. See also CALCULATE ROW and ONROWCALCS.	OFFROWCALCS
Re-enable calculations after using OFFROWCALCS. See also CALCULATE ROW and OFFROWCALCS.	ONROWCALCS
Define complex calculations for the row specified in CALCULATE ROW. SETROWOP defines a calculation operator to be applied to all subsequent output data rows. You can display the calculation results in the newly created row with the PRINTROW command.	SETROWOP
Immediately display the row specified in CALCULATE ROW to the report.	PRINTROW
Reset the value of the calculated row to #MISSING. See also CLEARALLROWCALC.	CLEARROWCALC
Reset the value of all calculated rows after using the CLEARROWCALC command.	CLEARALLROWCALC
Create a calculated row with captured data. See also SAVEANDOUTPUT.	SAVEROW
Capture data and output the result after using the SAVEROW command.	SAVEANDOUTPUT

For the syntax and definitions of row calculation commands, see the *Oracle Essbase Technical Reference*.

Commands that designate columns must use valid data column numbers, as determined by the *original* order of the columns.

- Precede and follow all operators in an expression with a single space.

- Essbase does not support nested (parenthetical) expressions.
- Essbase supports integer and floating-point constants in expressions as single entries or members of an array.

The CALCULATE ROW command can specify an operation (+, -, *, /, or OFF) as an equation consisting of constants, other calculated rows, and operators. Equations are evaluated at the time of declaration. Member names are not supported in expressions with the CALCULATE ROW command.

If you specify an operator, the operator applies to subsequent output rows and stores the result in the calculated row. Specifying an operator is useful for aggregating a series of rows to obtain a subtotal or total. To reset the operator, use SETROWOP. If the CALCULATE ROW command does not specify either an equation or an operator, the + operator is assumed.

The CALCULATE ROW command supports the standard mathematical operations. For syntax and parameter descriptions, see the *Oracle Essbase Technical Reference*.

This example is based on the Sample.Basic database.

```
{ CALC ROW "Total Sales" = "Sales..Group1"
     + "Sales..Group2" }
```

The example creates "Total Sales" based on two other calculated rows.

Changing How Data Is Displayed

Formatting commands customize how data displays in the final report:

Underlining

Use underlining as a visual aid to break up blocks of information in a report.

Task	Report Command
Set the default underline character to display in the report.	UNDERLINECHAR
Underline all characters that are not blank in the preceding row.	UCHARACTERS
Underline all the columns in the preceding row.	UCOLUMNS
Underline all the data columns for a row, while not underlining the row name columns.	UDATA
Underline all the row name columns in the preceding row while not underlining the data columns.	UNAME
Underline the row member names in a row whenever a member from the same dimension as the member in the command changes. Use the NOUNAMEONDIM command to turn off underlining for new rows.	UNAMEONDIMENSION

Suppressing Data Formatting

You can suppress data that you do not want to be displayed in the final report by using SUPPRESS commands.

Suppress	Report Command
Brackets around negative numbers. Use the BRACKETS command to re-enable brackets.	SUPBRACKETS
Commas in numbers greater than 999. Use the COMMAS command to re-enable commas.	SUPCOMMAS
Rows that have only zero data values. Use INCZEROROWS or INCEMPTYROWS to re-enable the display.	SUPZEROROWS
The European method for displaying numbers (2.000,01 whereas the non-European equivalent is 2,000.01). Use the EUROPEAN command to re-enable European number display.	SUPEUROPEAN
The automatic insertion of a page break whenever the number of lines on a page exceeds the current PAGELENGTH setting.	SUPFEED
Formats that produce output such as underlines and skips. Use INCFORMATS to re-enable the display.	SUPFORMATS
Text masks that were defined in the report using the MASK command. Use INCMASK to re-enable the display.	SUPMASK

See "Suppressing Page, Column, and Row Formatting" on page 525 for descriptions of the SUPPRESS commands used to suppress formats.

Indenting

Use indenting to provide visual clues to row levels of the script.

Task	Report Command
Shift the first-row names column in column output order by a specified number of characters.	INDENT
Indent subsequent row members in the row names column based on the generation in the database outline. Use the NOINDENTGEN command to left-justify row member names based on generation name.	INDENTGEN

Inserting Custom Titles

Titles are user-generated and optional, in contrast to the automatically generated page and column headings and row names, which describe the data on the report page.

Titles repeat at the top of each report page and provide the following information about a report:

- A date and time stamp
- The user name of the person running the report
- Page numbers
- The name of the source database

- Any other descriptive information

To add a title to the report, use the TEXT command, combined with any of the following:

- Any predefined keywords that automatically display information in the report
- A text string that you define

Note:

You can also use the TEXT command at the bottom of the report to provide summary information.

See the *Oracle Essbase Technical Reference* for the syntax and definitions of Report Writer commands.

Replacing Missing Text or Zeros with Labels

When you run a report, there are often many empty data cells where no data was applicable to the retrieval, or cells where the value is zero.

The report displays the default #MISSING label in the data cell when no data values are found.

➤ To replace the #MISSING label with a text label, at the point in the script where you want to replace the #MISSING label with a text label, type:

```
{MISSINGTEXT ["text"]}
```

where *text* is the text string that to be displayed in the data cells.

You can place the MISSINGTEXT command at any point in the report script; the command applies throughout the script.

Note:

You can also suppress #MISSING labels from appearing in the report. See "Suppressing Data Formatting" on page 531 for descriptions of SUPPRESS commands used to suppress labels, or see the *Oracle Essbase Technical Reference* for the syntax and definitions of Report Writer commands.

➤ To replace zeros with a text label, at the point in the script where you want to replace zeros with a text label, type

```
{ZEROTEXT ["text"]}
```

where *text* is the text string to be displayed in the data cells.

Note:

If a value is equal to #MISSING, any string inserted after that value using the AFTER command does not print. AFTER strings also do not print if you replace #MISSING with some other value (such as 0).

Adding Blank Spaces

Adding blank spaces in a report draws the reader to key information, such as totals.

Task	Report Command
Add one or more blank lines in the final report.	SKIP
Add a blank line when a member from the same dimension as the specified member in the command changes on the next line in the report. Use the NOSKIPONDIMENSION command to turn off insertion of a new line.	SKIPONDIMENSION

Changing How Data Values Display

You can use the following commands to change how data values display in the final report.

Task	Report Command
Turn on the display of commas for numbers greater than 999 after commas have been suppressed with either a SUPCOMMAS or SUPALL command.	COMMAS
Turn on the display of brackets around negative numbers instead of negative signs, after using the SUPBRACKETS command earlier in the script.	BRACKETS
Include a percentage sign or other character after the data values.	AFTER
Include a dollar sign or other character before the data values.	BEFORE
Use the European method for displaying numbers where decimal points are used as the thousands separator character and commas separate the decimal portion of the number from the integer portion (2.000,01; the non-European equivalent is 2,000.01).	EUROPEAN
Overwrite text in each output row with a specified characters and position.	MASK

Selecting and Sorting Members

The data that is displayed in the final report is based upon the members that you select and the order in which you display them. In addition, you can use conditional retrievals to further refine selecting and sorting members.

Selecting Members

Member selection commands are extraction commands that select ranges of members based on database outline relationships, such as sibling, generation, and level. Using member selection commands ensures that any changes to the outline are automatically reflected in the report, unless you change the member name on which the member selection command is based. Attribute dimensions can be included in member selection commands.

Task	Report Command
Select members from the same dimension as the dimension member.	ALLINSAMEDIM
Include siblings of the specified member.	ALLSIBLINGS
Include ancestors of the specified member.	ANCESTORS
Select a base dimension member based on its attributes.	ATTRIBUTE
Select members in the level immediately below the specified member.	CHILDREN
Include descendants of the specified member to the report, excluding the dimension top.	DESCENDANTS
Select level 0 members; that is, the members at the bottom of the dimension.	DIMBOTTOM
Include the top member of the dimension.	DIMTOP
Include a member and its ancestors.	IANCESTORS
Select the specified member and members in the level immediately below it.	ICHILDREN
Include the specified member and its descendants.	IDESCENDANTS
Include the specified member and its parent.	IPARENT
Include members from the same dimension and generation as the specified member.	OFSAMEGEN
Include members from the same dimension and on the same level as the specified member.	ONSAMELEVELAS
Include the parent of the specified member to the report.	PARENT
Extract data for a specified date or for a time period before or after a specific date.	TODATE
Include base dimension members associated with the specified attribute dimension.	WITHATTR

Selecting Members by Using Generation and Level Names

Generation and *level name* selection commands identify a specific level or generation of members based on either of the following items:

- The default generation or level name in the outline
- The user-defined generation or level name in the outline

When you use generation and level names, changes to the outline are automatically reflected in the report. You can define your own generation and level names or use the default names provided by Essbase. See "Generations and Levels" on page 65.

Using generation or level names whenever possible makes the report easier to maintain. Because you do not have to specify a member name in the report, you need not change the report if the member name is changed or deleted from the database outline.

Note:

Generation and level names are standalone commands. You cannot use them in place of member names in report extraction or formatting commands such as <DESCENDANTS or <CHILDREN.

➤ To use default level names, at the point in the script where you want to select a member by the default level name, use the format:

```
Levn,dimensionName
```

where *n* is the level number.

dimensionName is the name of the dimension from which you want to select the members.

Note:

Do *not* insert a space after the comma.

For example, Lev1,Year selects all the level 1 members of the Year dimension.

➤ To use default generation names, at the point in the script where you want to select a member by the default generation name, use the format:

```
Genn,dimensionName
```

where *n* is the generation number.

dimensionName is the name of the dimension from which you want to select the members.

Note:

Do *not* insert a space after the comma.

For example, Gen2,Year selects all the generation 2 members of the Year dimension.

Note:

These default generation and level names are not displayed in Outline Editor.

Based on the Sample.Basic database, the following example uses the default generation name Gen2,Year to generate a report that includes the members Qtr1, Qtr2, Qtr3, and Qtr4 from the Year dimension.

```
<PAGE(Product)
<COLUMN(Year)
<ROW (Measures)
{OUTALTNAMES}
Cola
Gen2,Year
```

```
Sales Profit
    !
```

Resulting report:

```
                          Cola Market Scenario
                    Qtr1      Qtr2      Qtr3      Qtr4
                  ======== ======== ======== ========
Sales               14,585    16,048    17,298    14,893
    Profit           5,096     5,892     6,583     5,206
```

Selecting Duplicate Members

The following Report Writer commands are available for use with duplicate member name outlines:

Task	Report Command
Displays member names for any unique member names and a qualified name for any duplicate member names.	REPQUALMBR
Displays member names only for members of the dimension specified.	REPMBR
Displays member names followed by aliases.	REPMBRALIAS
Displays alias names followed by member names for members in the report output.	REPALIAS
Displays aliases followed by member names for members of the dimension specified.	REPALIASMBR
Displays a member identifier for duplicate member names.	OUTPUTMEMBERKEY

Selecting Dynamic Time Series Members

You create and identify dynamic members in the database outline; they are members calculated only during user retrieval requests, such as generating a report script. The time dimension has a special Dynamic Time Series option. The Dynamic Time Series option has reserved the following generation names that you can define in the outline alias table:

Generation Name	Reserved Names	Explanation
History	H-T-D	History-to-Date
Year	Y-T-D	Year-to-Date
Season	S-T-D	Season-to-Date
Period	P-T-D	Period-to-Date
Quarter	Q-T-D	Quarter-to-Date
Month	M-T-D	Month-to-Date

Generation Name	Reserved Names	Explanation
Week	W-T-D	Week-to-Date
Day	D-T-D	Day-to-Date

See "Applying Predefined Generation Names to Dynamic Time Series Members" on page 442.

Note:

The database header message for the outline identifies the number of dynamic members that are enabled in the current outline.

➤ To select a Dynamic Time Series member, at the point in the script where you want to select a Dynamic Time Series member, use either:

- `<LATEST memberName`

  where *memberName* is the name of the member in the time dimension.

  The <LATEST command, a global command, is applied to the entire report script and is an aggregation based on the lowest level member within the dimension.

- `reservedName(memberName)`

  where *reservedName* is the reserved Dynamic Time Series generation name and the *memberName* is the name of the member in the time dimension.

  If you use this syntax to specify a Dynamic Time Series, the time series name is associated only to the member listed in the argument.

When you run the report script, the members are dynamically updated and the information is incorporated into the final report.

Note:

You must enter the Dynamic Time Series string exactly as it is displayed in the database outline; you cannot create a string and incorporate it into the final report. You can create an alias table for the Dynamic Time Series members in the database outline and use the aliases instead of the predefined generation names.

Selecting Members by Using Boolean Operators

Boolean operators enable you to specify precise member combinations within a report—useful when dealing with large outlines. Use the AND, OR, and NOT Boolean operators, combined with extraction commands, to refine member selections within the report script.

- AND, when all conditions must be met.
- OR, when one condition of several must be met.
- NOT, to choose the inverse of the selected condition.

➤ To create a Boolean expression using operators, at the point in the script where you want to use linking, enter the format:

```
<LINK (extractionCommand [operator extractionCommand])
```

where *extractionCommand* is the member selection command to retrieve data from, and *operator* is either the AND or OR operator.

Note:

Select members from the same dimension. All extraction command arguments must be enclosed in parentheses, as in the example above. NOT can be associated only with an extraction command and does not apply to the entire expression.

You can use Boolean operators with member selection commands, such as UDA and wildcards. See the *Oracle Essbase Technical Reference* for a list of all valid extraction commands that can be used with the LINK command.

Examples:

```
<LINK ((<IDESCENDANTS("100") AND <UDA(Product,Sweet)) OR ONSAMELEVELAS
"100"-10")
```

selects sweet products from the “100” subtree, plus all products on the same level as “100-10.”

```
<LINK ((<IDESCENDANTS("100") AND NOT <UDA (Product,Sweet)) OR ONSAMELEVELAS
"100"-10")
```

selects products that are not sweet from the “100” subtree, plus all products on the same level as “100-10.

See the *Oracle Essbase Technical Reference* for additional examples of narrowing member selection criteria.

Selecting Members by Using Substitution Variables

Substitution variables act as global placeholders for information that changes regularly; you set the substitution variables on the server through Administration Services, MaxL, or ESSCMD, and assign a value to each variable. You can then change the value anytime, reducing manual changes to a report script. You must have the role of at least Database Manager to set substitution variables.

For example, many reports are dependent on reporting periods; if you generate a report based on the current month, you must manually update the report script every month. With a substitution variable set on the server, such as CurMnth, you can change the assigned value each month to the appropriate time period. Essbase dynamically updates the information when you run the final report.

See “Using Substitution Variables” on page 120 for a comprehensive discussion about creating and changing substitution variables in the database outline. See the *Oracle Essbase Technical Reference* for information about the leading & character.

You can set substitution variables at the following levels:

- Server, providing access to the variable from all applications and databases on the server
- Application, providing access to the variable from all databases within the application
- Database, providing access to the specified database

➤ To use a substitution variable, at the point in the script where you want to use the variable, use the format:

```
&variableName
```

where *variableName* is the same as the substitution variable set on the server.

For example,

```
<ICHILDREN &CurQtr
```

becomes

```
<ICHILDREN Qtr1
```

Note:

The substitution variable must be accessible from the application and database against which you are running the report.

Note:

The variable name can be an alphanumeric combination whose maximum size is specified in Appendix A, "Limits." You cannot use spaces or punctuation in the variable name.

When you run the report script, Essbase replaces the variable name with the substitution value, and that information is incorporated into the final report.

Selecting Members by Using Attributes

Using attributes, you can select and report on data based on one or more characteristics of base members. You can group and analyze members of base dimensions according to their attributes. You can also perform crosstab reporting based on multiple attributes. Using the <ATTRIBUTE command, you can select all the base dimension members associated with an attribute. For example, you can query the Sample.Basic database on how many 12-ounce units of grape and orange juice were sold in New York during the first quarter.

➤ To select a member based on a specific attribute, at the point in the script where you want to select members based on a specific attribute, use the format:

```
<ATTRIBUTE memberName
```

where *memberName* is the name of an attribute-dimension member; for example:

```
<ATTRIBUTE Bottle
```

returns all products packaged in bottles.

Attribute dimensions have members with the same name. For example, the attribute dimension Ounces and the attribute dimension Age each can have a member named 24. To ensure that a query returns correct results, specify the full attribute-dimension member name. The following format returns all products that are packaged in 24 oz. units:

```
<ATTRIBUTE Ounces_24
```

Attribute types can be text, numeric, date, and Boolean. See "Understanding Attribute Types" on page 164.

Selecting Members by Attribute Association

You can select all base dimension members associated with one or more attributes using the <WITHATTR command. For example, you can display all products associated with a member of the Pkg Type attribute dimension. At the point in the script where you want to select the members, enter the following syntax:

```
<WITHATTR (attributeDimensionName, "Operator", Value)
```

The following command returns all base dimension members that are associated with the attribute Small from the Population attribute dimension.

```
<WITHATTR (Population, "IN", Small)
```

The following command returns all base dimension members that are associated with the attribute 32 from the Ounces attribute dimension.

```
<WITHATTR (Ounces, "<", 32)
```

Note:

The <WITHATTR command can be used within the LINK command to refine member selections, as illustrated in the following example: `<LINK ((<WITHATTR (Ounces, "<", 32) AND <WITHATTR ("Pkg Type", "=", Can))`

Selecting Members by Date

You can extract attributes data for a specific date, for a period before a specific date, or for a period after a specific date using the <TODATE command. For example, you can extract information on all products that were introduced on December 10, 1996, before December 10, 1996, or after December 10, 1996. The <TODATE command must be used within the <WITHATTR command. For example, the following format returns data on all products that were introduced *on* December 10, 1996.

```
<WITHATTR ("Intro Date", "=", <TODATE ("mm-dd-yyyy", "12-10-1996")
```

The following format returns data on all products that were introduced *before* December 10, 1996.

```
<WITHATTR ("Intro Date", "<", <TODATE ("mm-dd-yyyy", "12-10-1996")
```

The following format returns data on all products that were introduced *after* December 10, 1996.

```
<WITHATTR ("Intro Date", ">", <TODATE ("mm-dd-yyyy", "12-10-1996")
```

Note:

The types of date format supported are mm-dd-yyyy or dd-mm-yyyy. The date must be between January 1, 1970 and January 1, 2038 (inclusive).

Selecting Members by Using UDAs

UDAs enable you to select and report on data based on a common characteristic. UDAs are useful when performing member selections from an outline with an unbalanced hierarchy (in which the members of a dimension do not have identical member levels). You can set UDAs on the server for characteristics such as color, size, gender, flavor, or any other common member characteristics. You must have Database Manager permissions to set UDAs on the server.

UDAs are different from attributes. UDAs are member labels that you create to extract data based on a particular characteristic, but you cannot use UDAs to group data, to perform crosstab reporting, or to retrieve data selectively. For data analysis, UDAs are not as powerful as attributes.

You can use the UDA command with Boolean operators to refine report queries. See "Selecting Members by Using Boolean Operators" on page 537 for examples.

See "Creating UDAs" on page 157 for information about creating and maintaining UDAs.

➤ To select members based on a UDA, at the point in the script where you want to select members based on the UDA, use the format:

```
<UDA (dimensionName,"UDAstring")
```

where *dimensionName* is the dimension of the member that you select, and *UDAstring* is the UDA that is set on the server. The following example selects members of the Product dimension with the Sweet attribute.

```
<UDA (product,"Sweet")
```

When you run the report script, Essbase incorporates the UDA members into the final report.

Note:

You must type the UDA string exactly as it is displayed in the database outline; you cannot create a UDA string and incorporate it into the report script.

Selecting Members by Using Wildcards

You can use wildcards to select members, generation, or level names in a report script. If you use member names, Essbase searches the member and all descendants of that member. If you specify a generation or level name, Essbase searches only members of that generation or level.

Using wildcards reduces the member information needed for a script and simplifies script maintenance.

The following two types of wildcards are supported in Report Writer:

- Trailing asterisks (*) at the end of the string to search for common member properties
- Pattern-matching question marks (?) anywhere in the string to represent any single-character member property

➤ To select members using a trailing wildcard, at the point in the script where you want to select members using a trailing wildcard, use the format:

```
<MATCH (memberName,"character*")
```

where *memberName* is the name of the member that you select, and *character* is the beginning character in the following member. Using the Sample.Basic database,

```
<MATCH (Year,"J*")
```

returns Jan, Jun, and Jul.

➤ To select members using a pattern-matching wildcard, at the point in the script where you want to select members using a pattern-matching wildcard, use the format:

```
<MATCH (memberName,"???characters")
```

where *memberName* is the name of the member to select, and *characters* are the characters in the following member. Using the Sample.Basic database,

```
<MATCH (Product,"???-10")
```

returns 100-10, 200-10, 300-10, and 400-10.

Note:

In the Sample.Basic database example, three question marks represent the variable three characters in the string. If two question marks were used in the example, no matches were found. You can place question mark wildcards anywhere in the match string.

Suppressing Shared Members

In conjunction with the following items, you can suppress the display of later instances of shared members when you extract data for a report:

- Generation names
- Level names
- DIMBOTTOM command
- OFSAMEGEN command
- ONSAMELEVELAS command

Suppress shared members to eliminate unnecessary data duplication within the report.

➤ To suppress shared members, at the point in the script from which you want to suppress a shared member, use:

```
<SUPSHARE
```

<SUPSHARE suppresses the display of shared members for the duration of the report script. Use <SUPSHAREOFF to reset the display of shared members.

Selecting How Member Names are Displayed

Aliases make reports easier to read and help the reader focus on the data values rather than the meanings of member names. You can display members in a report by their aliases. For example, you can display page, column, and row names, such as Diet Cola or Caffeine Free Cola, rather than the corresponding member names 100-20 and 100-30.

Qualified member names help identify specific members in duplicate member databases. You can display the qualified member name when duplicate member names are encountered; for example, [State].[New York] and [City].[New York] uniquely identify the member name New York. For information about duplicate member names, see Chapter 8, "Creating and Working With Duplicate Member Outlines".

Task	Report Command
Display alias names for members of the specified dimension	<REPALIAS
Display alias names followed by member names	<REPALIASMBR
Display member names followed by alias names	<REPMBRALIAS
Display member names only for members of the dimension specified	<REPMBR
Display member names for unique member names and display qualified names for duplicate member names	<REPQUALMBR
Display member identifiers for duplicate member names in addition to member and alias name.	<OUTPUTMEMBERKEY

The first five commands in the table are format commands that can be applied to specific dimensions. They override one another. For example, when it encounters <REPALIASMBR Product, Report Writer displays the alias name in front of the member name. If <REPMBRALIAS Product is found later in the script, Report Writer then displays the member name in front of the alias name. For duplicate member outlines, the <OUTPUTMEMBERKEY command can be combined with any of the other commands in the table. The member identifier is included in the report output in addition to the member name, alias name, or both.

See the *Oracle Essbase Technical Reference* for command details.

Example Displaying a Member Name and Alias

You can display members in a report as a combination of the member name and its alias. Combining the member name and alias enables you to display more descriptive page, column, and row names, such as Diet Cola 100-20 or 100-30 Caffeine Free Cola.

➤ To display member names and aliases, use the <REPALIASMBR command or REPMBRALIAS command in a report script.

This example uses <REPALIASMBR to display the alias name ahead of the member name:

```
<PAGE (Product, Measures)
<COLUMN (Scenario, Year)
Actual
<ICHILDREN Qtr1
<ROW (Market)
<IDESCENDANTS "300"
<REPALIASMBR Product
    !
```

Resulting report:

```
                    Dark Cream 300-10 Measures Actual
                         Jan      Feb      Mar     Qtr1
                    ======== ======== ======== ========
Market                   800      864      880    2,544

                    Vanilla Cream 300-20 Measures Actual
                         Jan      Feb      Mar     Qtr1
                    ======== ======== ======== ========
Market                   220      231      239      690

                    Diet Cream 300-30 Measures Actual
                         Jan      Feb      Mar     Qtr1
                    ======== ======== ======== ========
Market                   897      902      896    2,695

                       Cream Soda 300 Measures Actual
                         Jan      Feb      Mar     Qtr1
                    ======== ======== ======== ========
Market                 1,917    1,997    2,015    5,929
```

Sorting Members

When you sort the members you include in a report, be aware that sorting commands affect members differently, depending on whether they are referenced by member selection commands or by static member definitions. Report Writer commands sort members by member name or data values.

Member selection commands such as <CHILDREN and <DESCENDANTS, select members in the order specified by the database outline. By default, a report that includes member selection commands displays members in their hierarchical database outline order. You can override this default by specifying a sort order with a sort command.

Because sort commands affect the order of the members selected by the member selection commands, they must precede any member selection commands to which they apply. If you specify a sort command, the sort order is preserved until another sort command overrides it.

Sort commands modify member selection commands, such as <CHILDREN and <DESCENDANTS. Sort commands do not perform final sorting of rows during formatting. Be

careful when you place a sort command in the report script that you do not start the sort too soon, and that you override it to turn it off, if necessary, before the next selection command.

Sort commands have no effect on static member definitions.

Task	Report Command
Sort members alphabetically by the alias name of the member, if aliases are used in the report script.	SORTALTNAMES
Sort following members in ascending order starting with the lowest generation and moving toward the highest generation.	SORTASC
Sort following members in descending order starting with the highest generation and moving toward the lowest generation.	SORTDESC
Sort following members according to the generation of the member in the database outline.	SORTGEN
Sort following members according to the level of the member in the database outline.	SORTLEVEL
Sort members alphabetically by member name.	SORTMBRNAMES
Disable all previous sorting commands so that members added to the report follow the normal hierarchical order based on the database outline.	SORTNONE

For a list of sorting commands syntax and descriptions, see the *Oracle Essbase Technical Reference*.

Restricting and Ordering Data Values

Several Report Writer commands let you perform conditional retrieval and data sorting in reports.

Task	Report Command
Specify the number of rows to return. These rows must contain the top values of a specific data column.	TOP
Specify the number of rows to return. These rows must contain the lowest values of a specific data column.	BOTTOM
Specify the conditions the columns of a data row must satisfy before the row is returned.	RESTRICT
Specify the ordering of the rows of a report, based on the data values of data columns.	ORDERBY

For the syntax, definitions, and detailed examples of these commands, see the *Oracle Essbase Technical Reference*.

Configurable variables are used during conditional retrievals. For information about setting the Report Writer configurable variables, see "Changing Buffer Size" on page 899.

Understanding the Order of Operation

<RESTRICT, <ORDERBY, <TOP, and <BOTTOM can be displayed anywhere in the report script and in any order. When using these commands, place all global script formatting commands before a Page member or a Column member, or before a <PAGE command or <COLUMN command that expands into Page or Column members (for example, IDESCENDANTS, or ICHILDREN).

Essbase extracts data and applies restrictions and ordering in the following order:

1. Applies RESTRICT and any existing restrictive option such as SUPPMISSING, SUPZEROS, and SUPEMPTYROWS.
2. Applies TOP or BOTTOM, or both.
3. Applies ORDERBY.

Essbase then returns rows and displays output.

Using TOP, BOTTOM, and ORDERBY with Sorting Commands

<TOP, <BOTTOM, and <ORDERBY commands sort the output of a report by its data values. Essbase applies <TOP and <BOTTOM first, followed by <ORDERBY. If the report contains a sort command, such as <SORTMBRNAMES, which sorts members and not data, Essbase applies the sort command first, followed by <TOP and <BOTTOM, and then <ORDERBY. <ORDERBY is the final sort.

Using RESTRICT

The arguments of the <RESTRICT command let you specify qualifications for selecting rows. Essbase includes only qualified rows in the resulting report output.

<RESTRICT works only on the range of rows that you specify in a row member selection.

Essbase processes the restrictions from left to right, and does not allow grouping with parentheses in the list of arguments.

For example, the following example is *not* a valid syntax:

```
RESTRICT (... (@DATACOL(1) > 300 AND @DATACOL(2) < 600)...)
```

Use only one <RESTRICT per report, as terminated by the **!** command. If a report script contains more than one report, each <RESTRICT overwrites the one in the previous report. For example:

```
RESTRICT (@DATACOL(1) > @DATACOL(2) AND 800 < @DATACOL(3)
OR @DATACOL(4) <> #MISSING)
```

This <RESTRICT command is equivalent in operation to the following syntax:

```
RESTRICT (((@DATACOL(1) > @DATACOL(2)) AND (800<@DATACOL(3))) OR (@DATACOL
(4) <> #MISSING))
```

Using ORDERBY

The <ORDERBY command orders the output rows according to the data values in the specified columns. Using an optional direction argument to the ORDERBY command, you can specify either an ascending order using the ASC flag, or descending order using the DESC flag. You can specify different sorting directions in different columns of the same report. If no direction argument is used, ascending (ASC) is the default order.

To determine the set of rows to be ordered, specify the row grouping dimension in the command. The default row grouping is the innermost row dimension.

Only one <ORDERBY is allowed per report, as terminated by the ! command. If a report script contains more than one report, each <ORDERBY overwrites the one in the previous report.

Using ORDERBY with Formatting Commands

<ORDERBY command guidelines:

- Avoid using row formatting commands when you are using <ORDERBY in a report. Formatting commands scheduled for a given point in the report may show up unexpectedly because <ORDERBY shifted the row that contained the member with formatting.
- In general, avoid using row formatting commands, and place overall script formatting before column members or commands that expand the column members (such as "ICHILDREN column dimension, <column ..., column member").

Using TOP and BOTTOM

The <TOP and <BOTTOM commands specify the qualified number of rows with the highest or lowest column values, respectively, within a row group to be returned in a report. If the row group member is not specified, the innermost row group dimension is the default row group.

You can use <TOP and <BOTTOM together in the same report, but only one <TOP and one <BOTTOM is allowed per report. In this case, the two commands should have the same data column as their argument in order to prevent confusion. The result of the <TOP and <BOTTOM command is sorted by the value of the data column specified in the command in descending order.

<TOP and <BOTTOM work only on the range of rows specified in row member selection.

Note:

If <TOP or <BOTTOM occurs with <ORDERBY, the ordering column of the <ORDERBY does not have to be the same as the data column of the <TOP or the <BOTTOM.

If any combination of the <ORDERBY, <TOP, or <BOTTOM commands exist together in a report script, the row group member (*<rowGroupMember>*) should be the same. This restriction removes any confusion about the sorting and ordering of rows within a row group.

Caution!

Essbase discards rows that contain #MISSING values in their sorting column from the set of extracted data rows before the applying the TOP or BOTTOM sort.

For example, this command returns two rows with the highest data values in col2 (Actual, Qtr2) per row group:

```
1- TOP (2, @DATACOL(2))
```

When you run this command against the Sample.Basic database, the row grouping is Product, which implies that for Florida, the report returns 100-10 and 100-30 product rows, and for Maine, the report returns 100-10, 100-40 product rows, and so on.

```
                              Actual              Budget
                         Qtr1        Qtr2      Qtr1     Qtr2
Florida     100-10        570         670       570      650
            100-20        235         345       321      432
            100-30        655         555       455      865
            100-40        342         342       432      234
Maine       100-10        600         800       800      750
            100-20        734         334       734      534
            100-30        324         321       235      278
            100-40        432         342       289      310
New York    100-10       1010        1210      1110      910
            100-20        960         760       650      870
            100-30        324         550       432      321
            100-40        880         980       880     1080
            100-50        #MI         #MI       #MI      #MI
```

This example returns rows with the highest data values in col2 (Actual, Qtr2) per report, because the row grouping is the "market."

```
2- TOP("market", 3, @DATACOL(2))
```

Resulting rows:

```
New York    100-10       1010      1210      1110      910
            100-40        880       980       880     1080
Maine       100-10        600       800       800      750
```

This example returns two rows with the lowest data values in col2 (Actual, Qtr2) per row group.

```
3- BOTTOM ("market", 2, @DATACOL(2))
```

Resulting rows:

```
Maine       100-20        734         334       734      534
            100-30        324         321       235      278
```

Note:

<TOP and <BOTTOM put an upper limit on the number of (qualified) rows returned after all restrictions are applied. This upper limit equals the number of rows in the <TOP plus the number of rows in the <BOTTOM commands.

Understanding How Other Report Configurations Affect TOP and BOTTOM

When using the <TOP and <BOTTOM commands, be aware that some other commands affect their operation. In particular, Essbase treats the <SUPPMISSING, <SUPZEROS, and <SUPEMPTYROWS options as restrictions and applies them to the extracted rows along with the <RESTRICT command restrictions. Essbase applies these optional restrictions to the data rows before processing the <TOP or <BOTTOM commands, and before applying an <ORDERBY command.

Using TOP and BOTTOM with Formatting Commands

Whenever a formatting command occurs in a report, it is appended to the member that follows it. For example, in this sequence, {UCOLUMNS}, the underline columns command is appended internally to the member that comes next. In Script 1, it is appended to the row member that can be described as "first child of market, assuming Florida."

```
SCRIPT 1                  SCRIPT 2
....                      ....
< ROW Market              {UCOL}
{UCOL }                   < Florida    (row member)
<ICHILDREN Market
< TOP ....                < BOTTOM ....
```

Script 2, appends {UCOLUMNS} to the row member Florida. Essbase executes {UCOLUMNS} whenever it encounters a row that has row member Florida. If the TOP or BOTTOM command returns a row that does not contain Florida, the formatting commands appended to the rows are never executed.

Therefore, it is a good idea to place all general formatting commands before a <COLUMN command, or a command that expands into column members to guarantee that the formatting is executed. However, do not use formatting commands that work on rows, because these rows may never be picked up by the <TOP or <BOTTOM command. Also avoid using <SAVEROW and <CALCULATE ROW with the <TOP and <BOTTOM commands.

Converting Data to a Different Currency

If the database has a currency partition, you can calculate currency conversions in report scripts. Use the <CURRENCY command to set the output currency and currency type. Use the <CURHEADING command to display the currency conversion heading.

Note:

Currency conversion is not supported across transparent partitions.

For information about creating a currency conversion application, see "Building Currency Conversion Applications and Performing Conversions" on page 203.

For the syntax and definitions of Report Writer commands, see the *Oracle Essbase Technical Reference*.

Generating Reports Using the C, Visual Basic, and Grid APIs

Task	C API Function	Visual Basic API Function	C Grid API Function
Start sending a report specification to the active database.	ESSBEGINREPORT	ESBBEGINREPORT	ESSGBEGINREPORT
Mark the end of a report specification being sent to the active database.	ESSENDREPORT	ESBENDREPORT	N/A
Send a report specification to the active database as a single string.	ESSREPORT	ESBREPORT	N/A
Send a report specification to the active database from a file.	ESSREPORTFILE	ESBREPORTFILE	ESSGREPORTFILE

See the *Oracle Essbase API Reference* for descriptions and syntax.

34

Writing MDX Queries

In This Chapter

Introduction

MDX, the data manipulation language for Essbase, is a joint specification of the XML for Analysis founding members. See http://www.xmla.org.

To complete the exercises in this chapter, which are based on the Sample.Basic database, use the MaxL Shell. Before continuing, start Essbase and log in to the MaxL Shell. Additionally, be prepared to use a text editor to create the sample queries as presented in this chapter.

Note:

You can use the MDX Script Editor in Administration Services Console instead of the MaxL Shell. However, the instructions in this chapter use the MaxL Shell.

Understanding Elements of a Query

In this section you will create a template to use as a basis for developing simple queries.

Most queries can be built upon the following grammatical framework:

```
SELECT
 {}
```

```
ON COLUMNS
FROM Sample.Basic
```

SELECT in line 1 is the keyword that begins the main body of all MDX statements.

The braces { } in line 2 are a placeholder for a *set*. In the above query, the set is empty, but the braces remain as a placeholder.

Exercise 1: Creating a Query Template

➤ To create a query template:

1 **Create a folder to store sample queries that can be run against the Sample.Basic database. For example, create a folder called "queries" under the** `ARBORPATH/app/Sample/Basic` **directory of the Essbase installation.**

2 **Using a text editor, type the following code into a blank file:**

```
SELECT {} ON COLUMNS FROM Sample.Basic
```

3 **Save the file as** `qry_blank.txt` **to your** `queries` **folder.**

Note:

If you are using the MDX Script Editor in Administration Services instead of a text editor, save the query as `qry_blank.MDX` from the editor instead.

Introduction to Sets and Tuples

A *set* is an ordered collection of one or more *tuples* that have the same dimensionality (see "Rules for Specifying Sets" on page 554 for an explanation of dimensionality).

A *tuple* is a way to refer to a member or a member combination from any number of dimensions. For example, in the Sample.Basic database, Jan is a tuple, as is (Jan, Sales), as is ([Jan],[Sales],[Cola],[Utah],[Actual]).

The member name can be specified in these ways:

1. By specifying the actual name or the alias; for example, `Cola`, `Actual`, `COGS`, and `[100]`

   If the member name starts with number or contains spaces, it should be within brackets; for example, `[100]`. Brackets are recommended for all member names, for clarity and code readability.

   For attribute members, the long name (qualified to uniquely identify the member) should be used; for example, `[Ounces_12]` instead of `[12]`.

2. By specifying dimension name or any one of the ancestor member names as a prefix to the member name; for example, `[Product].[100-10]` and `[Diet].[100-10]`. This is a recommended practice for all member names; it eliminates ambiguity and enables you to refer accurately to shared members.

Note:

Use no more than one ancestor in the member name qualification. Essbase returns an error if multiple ancestors are included. For example, `[Market].[New York]` is a valid name for New York, and so is `[East].[New York]`. However, `[Market].[East].[New York]` returns an error.

3. By specifying the name of a calculated member defined in the WITH section.

Recall that the braces { } in line 2 of your query template are a placeholder for a set. In this exercise, we will add a set to the query and run it.

Exercise 2: Running Your First Query

➤ To run the query:

1 **Open `qry_blank.txt`.**

2 **Because a set can be as simple as one tuple, add `Jan` as a set to the query template. Retain the braces (required for all set specifications except for sets that are produced by a function call).**

Type Jan inside the braces in line 2:

```
SELECT
 {Jan}
ON COLUMNS
FROM Sample.Basic
```

3 **Save the query as `ex2.txt`.**

4 **Ensure that Essbase is started (the `essbase.exe` process is running).**

5 **In order for Essbase to receive MDX statements, pass the statements to Essbase using either the MaxL Shell or MDX Script Editor in Administration Services. The examples in this chapter use the MaxL Shell.**

Start the MaxL Shell and log in with a valid user name and password. For example,

```
essmsh -l admin passwd
```

6 **Copy and paste the entire SELECT query into the MaxL Shell, but do not press Enter yet.**

7 **Enter a semicolon at the end, anywhere after Basic but before pressing Enter. The semicolon is not part of MDX syntax requirements, but it is required by MaxL Shell to indicate the end of a statement that is ready to execute.**

Note:

If you are using the MDX Script Editor in Administration Services, do not terminate with a semicolon.

8 **Press Enter. You should see results similar to the following.**

```
Jan
 8024
```

Rules for Specifying Sets

As described previously, a set is an ordered collection of one or more tuples.

For example, in the following query, {[100-10]} is a set consisting of one tuple.

```
SELECT
{[100-10]}
ON COLUMNS
FROM Sample.Basic
```

In the following query, {([100-10], [Actual])} is a also a set consisting of one tuple, though in this case, the tuple is not one member name. Rather, ([100-10], [Actual]) represents a tuple consisting of members from two dimensions, Product and Scenario.

```
SELECT
{([100-10], [Actual])}
ON COLUMNS
FROM Sample.Basic
```

When a set has more than one tuple, the following rule applies: In each tuple of the set, members must represent the same dimensions as do the members of other tuples of the set. Additionally, the dimensions must be represented in the same order. In other words, each tuple of the set must have the same *dimensionality*.

For example, the following set consists of two tuples of the same dimensionality.

```
{(West, Feb), (East, Mar)}
```

The following set breaks the dimensionality rule, because Feb and Sales are from different dimensions.

```
{(West, Feb), (East, Sales)}
```

The following set breaks the dimensionality rule, because although the two tuples contain the same dimensions, the order of dimensions is reversed in the second tuple.

```
{(West, Feb), (Mar, East)}
```

A set can also be a collection of sets, or it can be empty (containing no tuples).

A set must be enclosed in braces {} except in cases where the set is represented by an MDX function which returns a set.

Introduction to Axis Specifications

An axis is a specification determining the layout of query results from a database. Axes fit into MDX queries as follows:

```
SELECT <axis> [, <axis>...]
FROM <database>
```

At least one axis must be specified in any MDX query.

Up to 64 axes may be specified, beginning with AXIS(0) and continuing with AXIS(1)...AXIS (63). Using more than three axes is uncommon. The order of axes is not important; however, when a set of axes 0 through *n* are specified, no axis between 0 and *n* should be skipped. Additionally, a dimension cannot appear on more than one axis.

The first five axes have keyword aliases:

`ON COLUMNS`	can be used in place of	`AXIS(0)`
`ON ROWS`	may replace	`AXIS(1)`
`ON PAGES`	may replace	`AXIS(2)`
`ON CHAPTERS`	may replace	`AXIS(3)`
`ON SECTIONS`	may replace	`AXIS(4)`

For example, in the query

```
SELECT {Jan} ON COLUMNS FROM Sample.Basic
```

the axis specification is `{Jan}` `ON COLUMNS`.

Exercise 3: Running A Two-Axis Query

➤ To run a two-axis query:

1 **Open** `qry_blank.txt`**.**

2 **Add a placeholder for a second axis, by adding** `ON ROWS`**:**

```
SELECT
 {}
ON COLUMNS,
 {}
ON ROWS
FROM Sample.Basic
```

Note:

Remember to add the comma after ON COLUMNS.

3 **Save the new query template as** `qry_blank_2ax.txt`**.**

4 **As the set specification for the column axis, enter the Product members 100-10 and 100-20. These member names contain special characters, so you must use brackets. For example, add the text shown in bold:**

```
SELECT
 {[100-10],[100-20]}
ON COLUMNS,
 {}
ON ROWS
FROM Sample.Basic
```

Note:

The convention used here, to enclose all member names in brackets even if they do not contain special characters, is recommended.

5 As the set specification for the row axis, enter the Year members Qtr1 through Qtr4.

```
SELECT
 {[100-10],[100-20]}
ON COLUMNS,
 {[Qtr1],[Qtr2],[Qtr3],[Qtr4]}
ON ROWS
FROM Sample.Basic
```

6 Save the query as `ex3.txt`.

7 Paste the query into the MaxL Shell and run it, as described in Exercise 2: Running Your First Query.

Results:

	100-10	**100-20**
Qtr1	5096	1359
Qtr2	5892	1534
Qtr3	6583	1528
Qtr4	5206	1287

Exercise 4: Querying Multiple Dimensions on a Single Axis

➤ To query multiple dimensions on a single axis:

1 Open `qry_blank_2ax.txt`.

2 On the column axis, specify two tuples, each of which is a member combination rather than a single member. Enclose each tuple in parentheses, because more than one member is represented in each tuple.

```
SELECT
 {([100-10],[East]), ([100-20],[East])}
ON COLUMNS,
 {}
ON ROWS
FROM Sample.Basic
```

3 On the row axis, specify four two-member tuples, nesting each Quarter with Profit:

```
SELECT
 {([100-10],[East]), ([100-20],[East])}
ON COLUMNS,
 {
  ([Qtr1],[Profit]), ([Qtr2],[Profit]),
  ([Qtr3],[Profit]), ([Qtr4],[Profit])
 }
ON ROWS
FROM Sample.Basic
```

4 Save the query as `ex4.txt`.

5 Paste the query into the MaxL Shell and run it, as described in Exercise 2: Running Your First Query.

Results:

		100-10	100-20
		East	East
Qtr1	Profit	2461	212
Qtr2	Profit	2490	303
Qtr3	Profit	3298	312
Qtr4	Profit	2430	287

Cube Specification

A cube specification is the part of the query that determines which database is being queried. The cube specification fits into an MDX query as follows:

```
SELECT <axis> [, <axis>...]
FROM <database>
```

The <database> section follows the FROM keyword and should consist of delimited or nondelimited identifiers that specify first an application name and a then database name; for example, the following specifications are valid:

- `FROM Sample.Basic`
- `FROM [Sample.Basic]`
- `FROM [Sample].[Basic]`
- `FROM'Sample'.'Basic'`

Using Functions to Build Sets

Rather than creating sets member-by-member or tuple-by-tuple, you can use a function that returns a set. MDX includes several functions that return sets and several functions that return other values. For a complete reference of MDX functions supported by Essbase, see the MaxL section of the online *Oracle Essbase Technical Reference*.

Exercise 5: Using the MemberRange Function

The MemberRange function returns a range of members inclusive of and between two specified members of the same generation. Its syntax is as follows:

```
MemberRange (member1, member2, [,layertype])
```

where the first argument you provide is the member that begins the range, and the second argument is the member that ends the range. The *layertype* argument is optional. See the *Oracle Essbase Technical Reference*.

Note:

An alternate syntax for MemberRange is to use a colon between the two members, instead of using the function name: *member1* : *member2*.

➤ To use the MemberRange function:

1 **Open** `qry_blank.txt`**.**

2 **Delete the braces** `{}`**, which are unnecessary when you are using a function to return the set.**

3 **Use the colon operator to select a member range of Qtr1 through Qtr4:**

```
SELECT
 [Qtr1]:[Qtr4]ON COLUMNS
FROM Sample.Basic
```

4 **Paste the query into the MaxL Shell and run it, as described in Exercise 2: Running Your First Query.**

Qtr1, Qtr2, Qtr3, and Qtr4 are returned.

5 **Use the MemberRange function to select the same member range, Qtr1 through Qtr4.**

```
SELECT
 MemberRange([Qtr1],[Qtr4])
ON COLUMNS
FROM Sample.Basic
```

6 **Run the query. The same results should be returned.**

7 **Save the query as** `ex5.txt`**.**

Exercise 6: Using the CrossJoin Function

The CrossJoin function returns the cross product of two sets from different dimensions. Its syntax is as follows:

```
CrossJoin(set,set)
```

The CrossJoin function takes two sets from different dimensions as input and creates a set that is a cross product of the two input sets, useful for creating symmetric reports.

➤ To use the CrossJoin function:

1 **Open** `qry_blank_2ax.txt`**.**

2 **Replace the braces** `{}` **from the columns axis with** `CrossJoin()`**.**

```
SELECT
 CrossJoin ()
ON COLUMNS,
 {}
ON ROWS
FROM Sample.Basic
```

3 **Add two comma-separated pairs of braces as placeholders for the two set arguments you will provide to the CrossJoin function:**

```
SELECT
 CrossJoin ({}, {})
```

```
ON COLUMNS,
 {}
ON ROWS
FROM Sample.Basic
```

4 In the first set, specify the Product member `[100-10]`. In the second set, specify the Market members `[East]`, `[West]`, `[South]`, and `[Central]`.

```
SELECT
 CrossJoin ({[100-10]}, {[East],[West],[South],[Central]})
ON COLUMNS,
 {}
ON ROWS
FROM Sample.Basic
```

5 On the row axis, use CrossJoin to cross a set of Measures members with a set containing Qtr1:

```
SELECT
 CrossJoin ({[100-10]}, {[East],[West],[South],[Central]})
ON COLUMNS,
  CrossJoin (
    {[Sales],[COGS],[Margin %],[Profit %]}, {[Qtr1]}
  )
ON ROWS
FROM Sample.Basic
```

6 Save the query as `ex6.txt`.

7 Paste the query into the MaxL Shell and run it, as described in Exercise 2: Running Your First Query.

Results:

		100-10	**100-10**	**100-10**	**100-10**
		East	**West**	**South**	**Central**
Sales	**Qtr1**	5731	3493	2296	3425
COGS	**Qtr1**	1783	1428	1010	1460
Margin %	**Qtr1**	66.803	59.118	56.01	57.372
Profit %	**Qtr1**	45.82	29.974	32.448	24.613

When using CrossJoin, the order of arguments affects the order of tuples in the output.

Exercise 7: Using the Children Function

The Children function returns a set of all child members of the given member. Its syntax:

`Children (member)`

Note:

An alternate syntax for Children is to use it like an operator on the input member, as follows: *member.Children*. We will use the operator syntax in this exercise.

➤ To use the Children function to introduce a shortcut in the first axis specification:

1. **Open** `ex6.txt`**.**
2. **In the second set of the column axis specification, replace** `[East],[West],[South],[Central]` **with** `[Market].Children`**.**

   ```
   SELECT
    CrossJoin ({[100-10]}, {[Market].Children})
   ON COLUMNS,
     CrossJoin (
       {[Sales],[COGS],[Margin %],[Profit %]}, {[Qtr1]}
     )
   ON ROWS
   FROM Sample.Basic
   ```
3. **Save the query as** `ex7.txt`**.**
4. **Paste the query into the MaxL Shell and run it, as described in Exercise 2: Running Your First Query.**
5. **You should see the same results as were returned for Exercise 6: Using the CrossJoin Function.**

Working with Levels and Generations

In MDX, *layer* refers to generations and levels in an Essbase hierarchy.

In Essbase, generation numbers begin counting with 1 at the dimension name; higher generation numbers are those closest to leaf members in a hierarchy.

Level numbers begin with 0 at the leaf-most part of the hierarchy, and the highest level number is a dimension name.

A number of MDX functions take a layer you specify as an input argument and perform set operations based on the generation or level represented in the layer argument.

You can specify a layer argument in the following ways:

- Generation or level name; for example, `States` or `Regions`.
- The dimension name along with the generation or level name; for example, `Market.Regions` and `[Market].[States]`.
- The Levels function with a dimension and a level number as input. For example, `[Year].Levels(0)`.
- The Level function with a member as input. For example, `[Qtr1].Level` returns the level of quarters in Sample.Basic, which is level 1 of the Market dimension.
- The Generations function with a dimension and a generation number as input. For example, `[Year].Generations (3)`.
- The Generation function with a member as input. For example, `[Qtr1].Generation` returns the generation of quarters in Sample.Basic, which is generation 2 of the Market dimension.

Note:

In the Sample.Basic database, Qtr1 and Qtr4 are in the same layer. This means that Qtr1 and Qtr4 are also in the same generation. However, in a different database with a ragged hierarchy, Qtr1 and Qtr4 might not necessarily be in the same level, although they are in the same generation. For example, if the hierarchy of Qtr1 drills down to weeks, and the hierarchy of Qtr4 stops at months, Qtr1 is one level higher than Qtr4, but they are still in the same layer.

Exercise 8: Using the Members Function

The Members function can be used to return all members of a specified generation or level. Its syntax when used with a layer argument:

`Members` (layer)

where the *layer* argument you provide indicates the generation or level of members you want returned.

Note:

An alternate syntax for Members is *layer*.Members.

➤ To use the Members function:

1 **Open `qry_blank.txt`.**

2 **Delete the braces `{}`.**

3 **Use the Members function and the Levels function to select all level 0 members in the Market dimension of Sample.Basic:**

```
SELECT
  Members(Market.levels(0))ON COLUMNS
FROM Sample.Basic
```

4 **Paste the query into the MaxL Shell and run it, as described in Exercise 2: Running Your First Query.**

All states in the Market dimension are returned.

5 **Save the query as `ex8.txt`.**

Using a Slicer Axis to Set Query Point-of-View

A *slicer* axis is a way to limit a query to apply to only a specific area of the database. The optional slicer, if used, must be in the WHERE section of an MDX query. Furthermore, the WHERE section must be the last component of the query, following the cube specification (the FROM section):

```
SELECT {set}
ON axes
FROM database
WHERE slicer
```

Use the slicer axis to set the context of the query; it is usually the default context for all the other axes.

For example, for a query to select only Actual Sales in the Sample.Basic database, excluding budgeted sales, the WHERE clause might look like the following:

```
WHERE ([Actual], [Sales])
```

Because (Actual, Sales) is specified in the slicer axis, you need not include them in the ON AXIS (n) set specifications.

Exercise 9: Limiting the Results with a Slicer Axis

➤ To use the slicer axis to limit results:

1 **Open** `ex6.txt`**.**

2 **Paste the query into the MaxL Shell and run it, as described in Exercise 2: Running Your First Query.**

Note the results in one of the data cells; for example, notice that the first tuple, `([100-10],[East],[Sales],[Qtr1])`, has a value of 5731.

3 **Add a slicer axis to limit the data returned to budgeted values only.**

```
SELECT
 CrossJoin ({[100-10]}, {[East],[West],[South],[Central]})
ON COLUMNS,
  CrossJoin (
    {[Sales],[COGS],[Margin %],[Profit %]}, {[Qtr1]}
  )
ON ROWS
FROM Sample.Basic
WHERE (Budget)
```

4 **Paste the query into the MaxL Shell and run it.**

Note that the results are different.

5 **Save the query as** `ex9.txt`**.**

Common Relationship Functions

The following relationship functions return **sets** based on member relationships in the database outline:

Relationship Function	Description
Children	Returns the children of the input member.
Siblings	Returns the siblings of the input member.
Descendants	Returns the descendants of a member, with varying options.

The following relationship functions return a single member rather than a set:

Relationship Function	Description
Ancestor	Returns an ancestor at the specified layer.
Cousin	Returns a child member at the same position as a member from another ancestor.
Parent	Returns the parent of the input member.
FirstChild	Returns the first child of the input member.
LastChild	Returns the last child of the input member.
FirstSibling	Returns the first child of the input member's parent.
LastSibling	Returns the last child of the input member's parent.

For examples using relationship functions, see the MDX examples in the MaxL section of the *Oracle Essbase Technical Reference*.

Performing Set Operations

The following set functions operate on input sets without deriving further information from a database:

Pure Set Function	Description
CrossJoin	Returns a cross-section of two sets from different dimensions.
Distinct	Deletes duplicate tuples from a set.
Except	Returns a subset containing the differences between two sets.
Generate	An iterative function. For each tuple in *set1*, returns *set2*.
Head	Returns the first *n* members or tuples present in a set.
Intersect	Returns the intersection of two input sets.
Subset	Returns a subset from a set, in which the subset is a numerically specified range of tuples.
Tail	Returns the last *n* members or tuples present in a set.
Union	Returns the union of two input sets.

Exercise 10: Using the Intersect Function

The Intersect function returns the intersection two input sets, optionally retaining duplicates. Its syntax is as follows:

```
Intersect (set, set [,ALL])
```

Use the Intersect function to compare sets by finding tuples that are present in both sets.

➤ To use the Intersect function:

1 **Open** `qry_blank.txt`**.**

2 **Delete the braces** `{}` **from the axis, and replace them with** `Intersect ()`**.**

```
SELECT
 Intersect (

 )
ON COLUMNS
FROM Sample.Basic
```

3 **Add two comma-separated pairs of braces to use as placeholders for the two set arguments you will provide to the Intersect function:**

```
SELECT
 Intersect (
 { },
 { }
 )
ON COLUMNS
FROM Sample.Basic
```

4 **Specify children of East as the first set argument.**

```
SELECT
 Intersect (
 {  [East].children },
 { }
 )
ON COLUMNS
FROM Sample.Basic
```

5 **For the second set argument, specify all members of the Market dimension that have a UDA of "Major Market."**

Note:

For information about the UDA function, see the *Oracle Essbase Technical Reference*.

```
SELECT
 Intersect (
 {  [East].children },
 {  UDA([Market], "Major Market") }
 )
ON COLUMNS
FROM Sample.Basic
```

6 **Paste the query into the MaxL Shell and run it, as described in Exercise 2: Running Your First Query.**

The results include all children of East that have a UDA of "Major Market":

New York	Massachusetts	Florida
8202	6712	5029

7 **Save the query as** `ex10.txt`**.**

Exercise 11: Using the Union Function

The Union function joins two input sets, optionally retaining duplicates. Syntax:

`Union (set, set [,ALL])`

Use the Union function to lump two sets together into one set.

➤ To use the Union function:

1. **Open `ex10.txt`.**
2. **Replace Intersect with Union.**
3. **Paste the query into the MaxL Shell and run it.**

   If you compare the results with the results of the Intersect function in the previous exercise, you see that while Intersect returns a set containing only those children of East that have a UDA of "Major Market," Union returns {all children of east) + (all Market Members that have a UDA of "Major Market.")
4. **Save the query as `ex11.txt`.**

   For more examples using pure set-operative functions, see the *Oracle Essbase Technical Reference*.

Creating and Using Named Sets and Calculated Members

Calculated members and named sets are logical entities in query that can be used multiple times during the life of the query. Calculated members and named sets can save time in lines of code written as well as in execution time. The optional WITH section at the beginning of an MDX query is where you define the calculated members and/or named sets.

Calculated Members

A calculated member is a hypothetical member that exists for the duration of the query execution. Calculated members enable complex analysis without necessitating adding new members to the database outline. Calculated members are a storage place for calculation results performed on actual members.

Use the following guidelines for calculated member names:

- Associate the calculated member with a dimension; for example, to associated the member MyCalc with the Measures dimension, name it `[Measures].[MyCalc]`.
- Do not use actual member names to name calculated members; for example, do not name a calculated member `[Measures].[Sales]`, because Sales already exists in the Measures dimension.

Setting the solve order for each calculated member is recommended when you use multiple calculated members to create ratios or custom totals. For more information about solve order, see the MDX section of the *Oracle Essbase Technical Reference.*

Exercise 12: Creating a Calculated Member

This exercise includes the Max function, a common function for calculations. The Max function returns the maximum of values found in the tuples of a set. Its syntax is as follows:

```
Max (set, numeric_value)
```

➤ To create a calculated member:

1 Open `qry_blank_2ax.txt`.

2 On the row axis set, specify the children of Product.

```
SELECT
 {}
ON COLUMNS,
 {[Product].children}
ON ROWS
FROM Sample.Basic
```

3 At the beginning of the query, add a placeholder for the calculated member specification:

```
WITH MEMBER [].[]
 AS ''
SELECT
 {}
ON COLUMNS,
 {[Product].children}
ON ROWS
FROM Sample.Basic
```

4 To associate the calculated member with the Measures dimension and name it Max Qtr2 Sales, add this information to the calculated member specification:

```
WITH MEMBER [Measures].[Max Qtr2 Sales]
 AS ''
SELECT
 {}
ON COLUMNS,
 {[Product].children}
ON ROWS
FROM Sample.Basic
```

5 After the AS keyword and inside the single quotation marks, define the logic for the calculated member named Max Qtr2 Sales. Use the Max function with the set to evaluate (Qtr2) as the first argument, and the measure to evaluate (Sales) as the second argument.

```
WITH MEMBER [Measures].[Max Qtr2 Sales]
 AS '
  Max (
    {[Year].[Qtr2]},
    [Measures].[Sales]
  )'
SELECT
 {}
ON COLUMNS,
 {[Product].children}
ON ROWS
FROM Sample.Basic
```

6. The calculated member Max Qtr2 Sales is defined in the WITH section. To use it in a query, include it on one of the axes in the SELECT portion of the query. Select the predefined calculated member on the columns axis:

```
WITH MEMBER [Measures].[Max Qtr2 Sales]
 AS '
  Max (
    {[Year].[Qtr2]},
    [Measures].[Sales]
  )'
SELECT
 {[Measures].[Max Qtr2 Sales]}
ON COLUMNS,
 {[Product].children}
ON ROWS
FROM Sample.Basic
```

7. Paste the query into the MaxL Shell and run it, as described in Exercise 2: Running Your First Query.

   Results:

	Max Qtr2 Sales
100	27187
200	27401
300	25736
400	21355
Diet	26787

8. Save the query as `ex12.txt`.

Named Sets

You define named sets in the WITH section of the query. Doing so is useful because you can reference the set by name when building the SELECT section of the query. For example, the named set `Best5Prods` identifies a set of the five top-selling products in December:

```
WITH
SET [Best5Prods] AS
 'Topcount (
   [Product].members,
   5,
   ([Measures].[Sales], [Scenario].[Actual],
    [Year].[Dec])
  )'
SELECT [Best5Prods] ON AXIS(0),
{[Year].[Dec]} ON AXIS(1)
FROM Sample.Basic
```

Using Iterative Functions

The following functions loop through sets of data and perform search conditions and results that you specify.

Function	Description
Filter	Returns the subset of tuples in *set* for which the value of the search condition is TRUE.
IIF	Performs a conditional test and returns an appropriate numeric expression or set depending on whether the test evaluates to TRUE or FALSE.
Case	Performs conditional tests and returns the results you specify.
Generate	An iterative function. For each tuple in *set1*, returns *set2*.

Filter Function Example

The following query returns all Market dimension members for which the expression `IsChild ([Market].CurrentMember,[East])` returns `TRUE`; the query returns all children of East.

```
SELECT
 Filter([Market].Members,
        IsChild([Market].CurrentMember,[East])
 )
ON COLUMNS
FROM Sample.Basic
```

The Filter function in MDX is comparable to the RESTRICT command in Report Writer.

For more examples of Filter and other iterative functions, see the *Oracle Essbase Technical Reference*.

Working with Missing Data

When you are querying on a database, you can use the NON EMPTY keywords at the beginning of an axis specification to prevent cells containing no value from being included the result of the query.

The axis specification syntax including NON EMPTY:

```
<axis_specification> ::=
        [NON EMPTY] <set> ON
        COLUMNS | ROWS | PAGES | CHAPTERS |
       SECTIONS | AXIS (<unsigned_integer>)
```

Including the optional keywords NON EMPTY before the set specification in an axis causes suppression of slices in that axis that would contain entirely #MISSING values.

For any given tuple on an axis (such as `(Qtr1, Actual)`), a *slice* consists of the cells arising from combining this tuple with all tuples of all other axes. If all of these cell values are #MISSING, the NON EMPTY keyword causes elimination of the tuple.

For example, if even one value in a row is not empty, the entire row is returned. Including NON EMPTY at the beginning of the row axis specification would eliminate the following row slice from the set returned by a query:

Qtr1					
Actual	#MISSING	#MISSING	#MISSING	#MISSING	#MISSING

In addition to suppressing missing values with NON EMPTY, you can use the following MDX functions to handle #MISSING results:

- CoalesceEmpty, which searches numeric value expressions for non #MISSING values
- IsEmpty, which returns TRUE if the value of an input numeric-value-expression evaluates to #MISSING
- Avg, which omits missing values from averages unless you use the optional IncludeEmpty flag

For more information, see the MDX section of the *Oracle Essbase Technical Reference*.

Using Substitution Variables in MDX Queries

Substitution variables act as global placeholders for information that changes regularly; you set the substitution variables on the server through Administration Services, MaxL, or ESSCMD and assign a value to each variable. You can change the value anytime. You must have the role of at least Database Manager to set substitution variables. See "Using Substitution Variables" on page 120 for a comprehensive discussion on creating and changing substitution variables.

➤ To use a substitution variable in an MDX expression, consider:

- The substitution variable must be accessible from the application and database you are querying.
- A substitution variable has two components: the name and the value.
- The variable name can be an alphanumeric combination whose maximum size is specified in Appendix A, "Limits." Do not use spaces, punctuation, or brackets ([ ]) in substitution variable names used in MDX.
- At the point in the expression where you want to use the variable, show the variable name preceded by an ampersand (&); for example, where `CurMonth` is the name of the substitution variable set on the server, include `&CurMonth` in the MDX expression.
- When you perform the retrieval, Essbase replaces the variable name with the substitution value, and that value is used by the MDX expression.

For example, the expression is written showing the variable name `CurQtr` preceded with the &:

```
SELECT
 {[&CurQtr]}
ON COLUMNS
FROM Sample.Basic
```

When the expression is executed, the current value (Qtr1) is substituted for the variable name, and the expression that is executed is:

```
SELECT
 {[Qtr1]}
ON COLUMNS
FROM Sample.Basic
```

Querying for Properties

In MDX, *properties* describe certain characteristics of data and metadata. MDX enables you to write queries that use properties to retrieve and analyze data. Properties can be intrinsic or custom.

Intrinsic properties are defined for members in all dimensions. The intrinsic member properties defined for all members in an Essbase database outline are MEMBER_NAME, MEMBER_ALIAS, LEVEL_NUMBER, and GEN_NUMBER.

MDX in Essbase supports two types of custom properties: attribute properties and UDA properties. Attribute properties are defined by the attribute dimensions in an outline. In the Sample.Basic database, the Pkg Type attribute dimension describes the packaging characteristics of members in the Product dimension. This information can be queried in MDX using the property name [Pkg Type].

Attribute properties are defined only for specific dimensions and only for a specific level in each dimension. For example, in the Sample.Basic outline, `[Ounces]` is an attribute property defined only for members in the Product dimension, and this property has valid values only for the level 0 members of the Product dimension. The [Ounces] property does not exist for other dimensions, such as Market. The [Ounces] property for a non level 0 member in the Product dimension is a NULL value. The attribute properties in an outline are identified by the names of attribute dimensions in that outline.

The custom properties also include UDAs. For example, [Major Market] is a UDA property defined on Market dimension members. It returns a TRUE value if [Major Market] UDA is defined for a member, and FALSE otherwise.

Querying for Member Properties

Properties can be used inside an MDX query in two ways.

- You can list the dimension and property combinations for each axis set. When a query is executed, the specified property is evaluated for all members from the specified dimension and included in the result set.

  For example, on the column axis, the following query returns the GEN_NUMBER information for every Market dimension member. On the row axis, the query returns MEMBER_ALIAS information for every Product dimension member.

  ```
  SELECT
    [Market].Members
       DIMENSION PROPERTIES [Market].[GEN_NUMBER] on columns,
  ```

```
 Filter ([Product].Members, Sales > 5000)
    DIMENSION PROPERTIES [Product].[MEMBER_ALIAS] on rows
FROM Sample.Basic
```

When querying for member properties using the DIMENSION PROPERTIES section of an axis, a property can be identified by the dimension name and the name of the property, or by using the property name itself. When a property name is used by itself, that property information is returned for all members from all dimensions on that axis, for which that property applies. In the following query, the MEMBER_ALIAS property is evaluated on the row axis for Year and Product dimensions.

```
SELECT  [Market].Members
     DIMENSION PROPERTIES [Market].[GEN_NUMBER] on columns,
  CrossJoin([Product].Children, Year.Children)
     DIMENSION PROPERTIES [MEMBER_ALIAS] on rows
 FROM Sample.Basic
```

- Properties can be used inside value expressions in an MDX query. For example, you can filter a set based on a value expression that uses properties of members in the input set.

  The following query returns all caffeinated products that are packaged in cans.

```
SELECT
     Filter([Product].levels(0).members,
              [Product].CurrentMember.Caffeinated and
              [Product].CurrentMember.[Pkg Type] = "Can")
           Dimension Properties
           [Caffeinated], [Pkg Type] on columns
  FROM Sample.Basic
```

  The following query calculates the value [BudgetedExpenses] based on whether the current Market is a major market, using the UDA [Major Market].

```
WITH
  MEMBER [Measures].[BudgetedExpenses] AS
  'IIF([Market].CurrentMember.[Major Market],
   [Marketing] * 1.2, [Marketing])'

  SELECT  {[Measures].[BudgetedExpenses]} ON COLUMNS,
   [Market].Members ON ROWS
 FROM Sample.Basic
 WHERE ([Budget])
```

The Value Type of Properties

The value of an MDX property in Essbase can be a numeric, Boolean, or string type. MEMBER_NAME and MEMBER_ALIAS properties return string values. LEVEL_NUMBER, and GEN_NUMBER properties return numeric values.

The attribute properties return numeric, Boolean, or string values based on the attribute dimension type. For example, in Sample.Basic, the [Ounces] attribute property is a numeric property. The [Pkg Type] attribute property is a string property. The [Caffeinated] attribute property is a Boolean property.

Essbase allows attribute dimensions with date types. The date type properties are treated as numeric properties in MDX. When comparing these property values with dates, use the TODATE function to convert date strings to numeric before comparison.

The following query returns all Product dimension members that have been introduced on date 03/25/2007. Because the property [Intro Date] is a date type, the TODATE function must be used to convert the date string “03-25-2007” to a number before comparing it.

```
SELECT
 Filter ([Product].Members,
         [Product].CurrentMember.[Intro Date] =
         TODATE("mm-dd-yyyy","03-25-2007"))ON COLUMNS
FROM Sample.Basic
```

When a property is used in a value expression, you must use it appropriately based on its value type: string, numeric, or Boolean.

You can also query attribute dimensions with numeric ranges.

The following query retrieves Sales data for Small, Medium, and Large population ranges.

```
SELECT
 {Sales} ON COLUMNS,
 {Small, Medium, Large} ON ROWS
FROM Sample.Basic
```

When attributes are used as properties in a value expression, you can use range members to check whether a member's property value falls within a given range, using the IN operator.

For example, the following query returns all Market dimension members with the population range in Medium:

```
SELECT
  Filter(
       Market.Members, Market.CurrentMember.Population
       IN "Medium"
  )
ON AXIS(0)
FROM Sample.Basic
```

NULL Property Values

Not all members may have valid values for a given property name. For example, the MEMBER_ALIAS property returns an alternate name for a given member as defined in the outline; however, not all members may have aliases defined. In these cases A NULL value is be returned for those members that do not have aliases.

In the following query,

```
SELECT
  [Year].Members
   DIMENSION PROPERTIES [MEMBER_ALIAS]
ON COLUMNS
FROM Sample.Basic
```

none of the members in the Year dimension have aliases defined for them. Therefore, the query returns NULL values for the MEMBER_ALIAS property for members in the Year dimension.

The attribute properties are defined for members of a specific dimension and a specific level in that dimension. In the Sample.Basic database, the [Ounces] property is defined only for level 0 members of the Product dimension.

Therefore, if you query for the [Ounces] property of a member from the Market dimension, as shown in the following query, you will get a syntax error:

```
SELECT
    Filter([Market].members,
           [Market].CurrentMember.[Ounces] = 32) ON COLUMNS
FROM Sample.Basic
```

Additionally, if you query for the [Ounces] property of a non level 0 member of the dimension, you will get a NULL value.

When using property values in value expressions, you can use the function IsValid() to check for NULL values. The following query returns all Product dimension members with an [Ounces] property value of 12, after eliminating members with NULL values.

```
SELECT
  Filter([Product].Members,
         IsValid([Product].CurrentMember.[Ounces]) AND
         [Product].CurrentMember.[Ounces] = 12)
ON COLUMNS
FROM Sample.Basic
```

35 Copying Data Subsets and Exporting Data to Other Programs

In This Chapter

Process for Creating a Database Subset

You can move data between Essbase databases or to other programs by extracting an output file of the data that you want to move. To meet the import format specifications of most other programs, create a text file using a report script or a calculation script.

This section describes the process for copying a database subset from one Essbase Server (Essbase Server 1) to another (Essbase Server 2).

1. On Essbase Server 2, create an application and database to contain the database subset.

   See "Create an Application and Database" on page 576.

2. Copy the outline file (for example, `source_dbname.otl`) from the source database on Essbase Server 1 to the new database on Essbase Server 2.

   You may need to rename the outline file to match the name of the database Essbase Server 2 (for example, `target_dbname.otl`), overwriting the existing target database outline file.

   See "Copy the Outline File from the Source Database" on page 576.

3. Create an output file (for example, a plain text file) containing the required data subset.

   See "Create an Output File Containing the Required Data Subset" on page 577.

4. Load the output file into the new database that you have created.

   See "Load the Output File into the New Database" on page 578.

If required, repeat steps 3 and 4 to create an output file from the database on Essbase Server 2 and load the data back into the main Essbase database on a different computer.

The example in the following sections is based on the Sample.Basic database. The data subset in the example is the Actual, Measures data for the West market. The example copies the data subset to Essbase Server 2 and the West Westmkts database.

Create an Application and Database

Create an application and database on Essbase Server 2. You will copy the required subset of data into this new database. You can give this application and database any name.

➤ To create the application and database, see "Creating Applications" and "Creating Databases" in the *Oracle Essbase Administration Services Online Help*.

Ensure that the new, empty database is not running.

➤ To stop a database, see "Stopping Databases" in the *Oracle Essbase Administration Services Online Help*.

Copy the Outline File from the Source Database

Copy the outline file (`.otl`) of the source database to the new database that you have created. In this example, you copy the `basic.otl` outline file from the Sample.Basic database and rename it `wesmkts.otl` on Essbase Server 2.

How you copy the outline file depends on whether you can connect to the source Essbase database from the Essbase Server 2 computer.

- If you can connect, use any of the following methods to copy the outline:

Tool	Topic	Location
Administration Services	Copying Outlines	*Oracle Essbase Administration Services Online Help*
MaxL	**create database**	*Oracle Essbase Technical Reference*
ESSCMD	COPYDB	*Oracle Essbase Technical Reference*

- If you cannot connect, use the operating system to copy the outline file.
  1. Use the operating system to copy the source outline file; for example, copy `basic.otl` to `westmkts.otl`.
  2. Give the copied outline exactly the same name as the new database.
  3. Save the outline file in the *`ARBORPATH`*`/app/`*`appname`*`/`*`dbname`* directory on the Essbase Server 2 computer, where *`ARBORPATH`* is the directory in which you installed Essbase, and *`appname`* and *`dbname`* are the new application and database that you have created.

     For example, copy `basic.otl` to a disk, renaming it to `westmkts.otl`. Then copy `westmkts.otl` from the disk to `Hyperion/products/Essbase/EssbaseServer/app/west/westmkts/westmkts.otl` on the Essbase Server 2 computer. It is safe to overwrite the existing, empty `westmkts.otl` file.

     Note:

     Ensure that the new outline file overwrites the existing, empty outline file, which Essbase created automatically when you created the new application and database.

4. Stop and restart the new database.

   See “Starting Databases” and “Stopping Databases” in the *Oracle Essbase Administration Services Online Help*.

You now have a copy of the database outline on Essbase Server 2.

Create an Output File Containing the Required Data Subset

Create an output file that contains the required data subset. The output file can be a text file or a spreadsheet file. Use either of the following methods to create a data subset.

➤ To create an output file for a subset of a database, use a tool:

The following example uses Report Writer to create a subset of the Westmkts database. You can also use a report script to export a subset of data.

➤ To create a text file that contains the required data subset:

1 **Select the source database. For example, select West Westmkts.**

See “Navigating and Selecting Objects” in the *Oracle Essbase Administration Services Online Help*.

- If you *can* connect to the Essbase Server 1 database from the Essbase Server 2 computer, you can select the source database from Essbase Server 2.
- If you *cannot* connect, use a different computer from the Essbase Server 2 computer to select the source database.

2 **Create a report.**

See “Creating Scripts” in the *Oracle Essbase Administration Services Online Help*.

3 **Write a report script that selects the required data subset. For information about writing report scripts, see Chapter 32, “Understanding Report Script Basics.”**

For example, the following report script selects the Actual, Measures data for the West market from Sample.Basic:

```
{TABDELIMT}
<QUOTEMBRNAMES
Actual
<IDESC West
<IDESC Measures
```

- Use TABDELIMIT to place tab stops between data, instead of spaces, to ensure that no member names or data values are truncated.
- Use QUOTEMBRNAMES to place quotation marks (" ") around member names that contain blank spaces. Essbase then recognizes the member names when it loads the data.

4 **Execute the report script.**

See “Executing Report Scripts” in the *Oracle Essbase Administration Services Online Help*.

5 **Save the report script with a `.txt` extension; for example, `westout.txt`.**

To load the data, the output file must be in the *`ARBORPATH`*`/app/`*`appname`*`/`*`dbname`* directory on Essbase Server 2, where *`appname`* and *`dbname`* are the new application and database directories that you have created.

If you are using the Essbase Server 2 computer, you can save the output file directly into the *`ARBORPATH`*`/app/`*`appname`*`/`*`dbname`* directory; for example:

```
Hyperion/products/Essbase/EssbaseServer/app/west/westmkts/westout.txt
```

If you are not using the Essbase Server 2 computer, save the output file anywhere on the current computer. By default, Essbase saves the file on the Essbase client computer and not on the server. When you run the report, use the operating system to copy the file to the *`ARBORPATH`*`/app/`*`appname`*`/`*`dbname`* directory on Essbase Server 2. For example, use a disk to copy the file.

If you are not using the Essbase Server 2 computer, download and copy the file from the Essbase Server client directory to the *`ARBORPATH`*`/app/`*`appname`*`/`*`dbname`* directory on Essbase Server 2. For example, copy the output file to:

```
Hyperion/products/Essbase/EssbaseServer/app/west/westmkts/westout.txt
```

You are now ready to load the text file into the new database.

Load the Output File into the New Database

Load the output file into the new database on Essbase Server 2.

➤ To load a file into a database, see "Performing a Data Load or Dimension Build" in the *Oracle Essbase Administration Services Online Help*.

The following example illustrates how to load data into the Westmkts database:

1. Select the new database. For example, select `Westmkts`.
2. Start the data load using the text file you have just created, for example, `westout`.

Note:

If `westout` is not displayed, check that you gave it a `.txt` extension and placed it in the *ARBORPATH*`/app/West/Westmkts` directory. See "Create an Output File Containing the Required Data Subset" on page 577.

See Chapter 19, "Performing and Debugging Data Loads or Dimension Builds."

You can now view the data on the Essbase Server 2 computer. You might need to recalculate the database subset. Because you are viewing a subset of the database, a percentage of data values will be #MISSING.

If required, you can copy report scripts and other artifact files to the Essbase Server 2 computer to use with the database subset that you have created.

Exporting Data to be Input to Other Programs

➤ To export data from a database to a format that can be read by programs other than Essbase, use a tool:

Tool	Topic	Location
Spreadsheet Add-in Query Designer	Applying Queries	*Oracle Essbase Spreadsheet Add-in User's Guide*
Smart View	"Working with Query Designer"	*Oracle Hyperion Smart View for Office User's Guide*
Calculation scripts	"Using DATAEXPORT to Export Data" on page 461	*Oracle Essbase Database Administrator's Guide*
Report scripts	"Exporting Text Data Using Report Scripts" on page 579	*Oracle Essbase Database Administrator's Guide*

Note:

When you use MaxL, calculation scripts, or Administration Services Console to export text data, and the end fields of the last record have no values, those fields will not contain the missing value marker (such as #MI). To improve overall processing time, the fields are left blank in the last output record.

Export files from databases in Unicode-mode applications are in UTF-8 encoding. If Essbase finds insufficient disk space for the export file, no data is exported.

If you plan to import Essbase data into a program that requires special delimiters, use the Report Writer MASK command or the delimiter parameter of the DATAEXPORT calculation command to specify the delimiter.

For more information about exporting data for backup purposes, see the *Oracle Hyperion Enterprise Performance Management System Backup and Recovery Guide*.

You can use report scripts or calculation scripts to export Essbase data in text format. Both tools enable you to create text files that meet the import format specifications of most other programs.

Exporting Text Data Using Report Scripts

Report Writer provides extensive flexibility in selecting output data and formatting it to meet the needs of other programs. Report scripts work with all member types (for example: stored members, Dynamic Calc members, attributes, label-only members, shared members, and aliases.) However, report scripts generally are slower because they use query-based data extraction, which probes data blocks that are not materialized in the database. See "Executing Report Scripts" in the *Oracle Essbase Administration Services Online Help* and "Executing Report Scripts" on page 512. Report script-based export provides more flexibility in formatting the data and is ideal for generating professional-looking reports.

When you export data to a program that uses a two-dimensional, fixed-field format, you need not specify page or column dimensions. To create a two-dimensional report, you can specify every dimension as a row dimension. Use the ROWREPEAT command to add the name of each member specified to each row (rather than the default, nested style). The following script example and report illustrate this situation for a five-dimensional database:

```
<ROW (Year, Measures, Product, Market, Scenario)
{ROWREPEAT}
<ICHILDREN Year
Sales
<ICHILDREN "400"
East
Budget
    !
```

Resulting report:

```
Qtr1          Sales        400-10       East      Budget      900
Qtr1          Sales        400-20       East      Budget    1,100
Qtr1          Sales        400-30       East      Budget      800
Qtr1          Sales          400        East      Budget    2,800
Qtr2          Sales        400-10       East      Budget    1,100
Qtr2          Sales        400-20       East      Budget    1,200
Qtr2          Sales        400-30       East      Budget      900
Qtr2          Sales          400        East      Budget    3,200
Qtr3          Sales        400-10       East      Budget    1,200
Qtr3          Sales        400-20       East      Budget    1,100
Qtr3          Sales        400-30       East      Budget      900
Qtr3          Sales          400        East      Budget    3,200
Qtr4          Sales        400-10       East      Budget    1,000
Qtr4          Sales        400-20       East      Budget    1,200
Qtr4          Sales        400-30       East      Budget      600
Qtr4          Sales          400        East      Budget    2,800
  Year        Sales        400-10       East      Budget    4,200
  Year        Sales        400-20       East      Budget    4,600
  Year        Sales        400-30       East      Budget    3,200
  Year        Sales          400        East      Budget   12,000
```

To create a two-dimensional report that contains only bottom-level (level 0) data, use CHILDREN or DIMBOTTOM to select level 0 members.

- To list only level 0 data for specific members, use the CHILDREN command with the level 1 member as a parameter above the data that you want to print.
- To list all level 0 data for the dimension to which a given member belongs, use the DIMBOTTOM command with any member in the dimension that contains the data that you want to print.

For example, the following script uses the CHILDREN command to select the children of Qtr1, which is a level 1 member, and the DIMBOTTOM command to select all level 0 data in the Product dimension.

```
<ROW (Year, Measures, Product, Market, Scenario)
{ROWREPEAT}
{DECIMAL 2}
<CHILDREN Qtr1
Sales
```

```
<DIMBOTTOM Product
East
Budget
     !
```

Resulting report:

```
Jan      Sales    100-10     East       Budget         1,600.00
Jan      Sales    100-20     East       Budget           400.00
Jan      Sales    100-30     East       Budget           200.00
Jan      Sales    200-10     East       Budget           300.00
Jan      Sales    200-20     East       Budget           200.00
Jan      Sales    200-30     East       Budget         #Missing
Jan      Sales    200-40     East       Budget           700.00
Jan      Sales    300-10     East       Budget         #Missing
Jan      Sales    300-20     East       Budget           400.00
Jan      Sales    300-30     East       Budget           300.00
Jan      Sales    400-10     East       Budget           300.00
Jan      Sales    400-20     East       Budget           400.00
Jan      Sales    400-30     East       Budget           200.00
Feb      Sales    100-10     East       Budget         1,400.00
Feb      Sales    100-20     East       Budget           300.00
Feb      Sales    100-30     East       Budget           300.00
Feb      Sales    200-10     East       Budget           400.00
Feb      Sales    200-20     East       Budget           200.00
Feb      Sales    200-30     East       Budget         #Missing
Feb      Sales    200-40     East       Budget           700.00
Feb      Sales    300-10     East       Budget         #Missing
Feb      Sales    300-20     East       Budget           400.00
Feb      Sales    300-30     East       Budget           300.00
Feb      Sales    400-10     East       Budget           300.00
Feb      Sales    400-20     East       Budget           300.00
Feb      Sales    400-30     East       Budget           300.00
Mar      Sales    100-10     East       Budget         1,600.00
Mar      Sales    100-20     East       Budget           300.00
Mar      Sales    100-30     East       Budget           400.00
Mar      Sales    200-10     East       Budget           400.00
Mar      Sales    200-20     East       Budget           200.00
Mar      Sales    200-30     East       Budget         #Missing
Mar      Sales    200-40     East       Budget           600.00
Mar      Sales    300-10     East       Budget         #Missing
Mar      Sales    300-20     East       Budget           400.00
Mar      Sales    300-30     East       Budget           300.00
Mar      Sales    400-10     East       Budget           300.00
Mar      Sales    400-20     East       Budget           400.00
Mar      Sales    400-30     East       Budget           300.00
```

For another example of formatting for data export, see “Sample 12 on the Examples of Report Scripts” page in the “Report Writer Commands” section of the *Oracle Essbase Technical Reference*.

Exporting Text Data Using Calculation Scripts

You can use the following calculation commands to select and format a text import file: DATAEXPORT, DATAEXPORTCOND, SET DATAEXPORTOPTIONS, FIX…ENDFIX, and

EXCLUDE...ENDEXCLUDE.. For general information about creating and running a calculation script, see Chapter 28, “Developing Calculation Scripts.”

Calculation script-based data export works with stored and dynamically calculated members only and provides fewer formatting options than report scripts. However, calculation script-based data exports provide decimal- and precision-based formatting options and can be faster than report scripts. The DATAEXPORT calculation command also enables export directly to relational databases, eliminating the usual intermediate import step.

The following calculation script example produces a text file that contains a subset of the database.

```
SET DATAEXPORTOPTIONS
{   DATAEXPORTLEVEL "ALL";
DATAEXPORTCOLFORMAT ON;
DATAEXPORTCOLHEADER Scenario;
};
FIX ("100-10","New York","Actual","Qtr1");
   DATAEXPORT "File" "," "C:\exports\actual.txt" "NULL";
ENDFIX;
```

These commands specify inclusion of all levels of data and indicate that data is to be repeated in columns, with the Scenario dimension set as the dense dimension column header for the output. The FIX command defines the data slice, and then the data is exported to a text file at `C:\exports\actual.txt`. Commas are used as delimiters, and missing data values are indicated by consecutive delimiters. Running this script against Sample.Basic generates the following data:

```
"Actual"
"100-10","New York","Sales","Qtr1",1998
"100-10","New York","COGS","Qtr1",799
"100-10","New York","Margin","Qtr1",1199
"100-10","New York","Marketing","Qtr1",278
"100-10","New York","Payroll","Qtr1",153
"100-10","New York","Misc","Qtr1",2
"100-10","New York","Total Expenses","Qtr1",433
"100-10","New York","Profit","Qtr1",766
"100-10","New York","Opening Inventory","Qtr1",2101
"100-10","New York","Additions","Qtr1",2005
"100-10","New York","Ending Inventory","Qtr1",2108
"100-10","New York","Margin %","Qtr1",60.01001001001001
"100-10","New York","Profit %","Qtr1",38.33833833833834
"100-10","New York","Profit per Ounce","Qtr1",63.83333333333334
```

For information about DATAEXPORT calculation commands, see the *Oracle Essbase Technical Reference*.

36 Mining an Essbase Database

In This Chapter

Understanding Data Mining

Data mining tools can sift through data to come up with hidden relationships and patterns. You may find that people who bought root beer in July also bought ice cream in July. Then you can use this knowledge to create an advertising campaign around root beer floats.

You can use data mining to tell you things about existing data, as in the root beer example. Such data mining is called descriptive. You can also use data mining to forecast future results based on past performance. For example, you can forecast sales for next year based on sales for this year. Such data mining is called predictive.

The data mining process involves using algorithms to analyze a data and then, from the relationships that it finds, the algorithm results are projected against another set of data. An *algorithm* is a method (set of instructions) that is used to analyze the data and apply the collected knowledge to other data.

For example, suppose you want to determine how sales of televisions, DVDs, and VCRs in the East region correspond to sales of cameras in the same region. You can use the regression algorithm to model sales of cameras as a function of sales of TVs, VCRs, and DVDs. The algorithm performs a series of mathematical steps to create a concise representation (model) of the knowledge obtained from analysis of data. Then it uses this model to predict the value of a dependent (target) variable based on the value of one or more independent (predictor) variables.

In some cases, in order to better fit model to given data, it may be necessary to modify the data. *Transformations* are used to mathematically adjust data values before an algorithm analyzes the data. For example, a transformation could cause an exponentiation of values that the algorithm would include in its analysis.

Essbase Data Mining Framework

Data Mining Framework is a collection of features that enables the execution of data mining on Essbase databases. Data Mining Framework is licensed separately from Essbase.

Note:

Data Mining Framework does not currently operate on relational data, that is, Hybrid Analysis data. Data mining is also not supported currently for Unicode-mode applications. The names of data mining algorithms, models, transformations and templates must contain ASCII characters only.

To understand the processes for mining an Essbase database you need to become familiar with the following terms:

- Algorithm: a method (set of instructions) that is used to analyze the data.
- Model: a collection of an algorithm's findings about examined data. A model can be used (applied) against a different set of data to generate useful information about that data.
- Task: a step in the data mining process.
- Task template: a task specification that can be modified and used again for a variation of that task.

Essbase Data Mining Tasks

The Essbase data mining procedure includes the following tasks:

- Specify a build task.

  You implement a build task through a task template. The Mining Wizard, available with Administration Services, provides a build task template and steps you through the process of specifying a build task. You can also use MaxL statements and sample template files provided with Data Mining Framework.

  In the build task, you specify:

  - The algorithm to use.
  - The database to mine and a representative set of data.
  - Parameters that determine how the algorithm is applied.
- Execute the build task to build, or train, a model.

  When you execute a build task, the specified algorithm is applied against the specified set of data to generate a data mining model. During the training process, the algorithm discovers and describes internally the patterns and relationships in the data that can be used for prediction. Later, the trained model can use these patterns and relationships to generate new information from a different, but similarly structured, set of data.

  See “Training the Model” on page 588.

- Specify a test task to test the model. This is an optional task to verify the accuracy of the model you have built.

  You implement a test task through a task template using the Mining Wizard or MaxL statements.

  In the test task, you specify:

  - The model to use. This is a model that you have built.
  - A database and set of data to mine. Specify a set of data with known results or correlations.
- Execute the test task to generate test results.

  After training the model on one set of data, you execute the model against a different set of data with known results or correlations. You can compare the results generated by the model with the known results to determine the accuracy of the model.

  See "Testing the Model" on page 589.
- Specify an apply task.

  You implement an apply task through a task template using the Mining Wizard or MaxL statements.

  In the apply task, you specify:

  - The model to use. This is a model that you have built.
  - A database and set of data to mine
  - Where in the cube to store the results
- Execute the apply task to generate result records.

  When you execute an apply task, the model is executed against the specified set of data to generate result records. Data Mining Framework writes results back to the Essbase cube.

  See "Applying the Model" on page 589.
- Query the models or mining results.

  Query the models or results to view their contents.
- Specify a scoring task.

  You implement an scoring task through a task template using the Mining Wizard or MaxL statements.

  In the scoring task, you specify:

  - The model to use. This is a model that you have built.
  - The data values to be mined. These values may come from the Essbase cube (cube scoring) or may be entered manually (data scoring).
- Execute the scoring task to generate results.

  When you execute a scoring task, the model is executed against the data you specify and the results are displayed by the Administration Services Console.

The following sections describe the data mining process in more detail.

Creating Data Mining Models

The first step in building a data mining model is to understand the business situation or problem and choose the appropriate algorithm to use for analysis or prediction. Algorithms determine how you process data.

Data Mining Framework provides a set of powerful algorithms that can be used for a broad range of business applications. See "Algorithms" on page 591 for a description of the available algorithms. The description of each algorithm provides information about the type of problem that the algorithm is best suited for.

Note:

Data Mining Framework also enables you to easily register and use new algorithms created by Oracle or third-party vendors.

Preparing for Data Mining

The one essential prerequisite for performing data mining is that you understand your data and the problem you are trying to solve. Data mining is a powerful tool and can yield new insights. If you already have a strong hunch about your data, data mining can be particularly useful in confirming or denying your hunch, and giving you some additional insights and directions to follow.

Before you mine an Essbase database, make sure that the database is loaded and calculated.

Using an Algorithm

All data mining algorithms require training, or learning. Training is the process of executing an algorithm against a representative set of data to discover and describe the patterns and relationships in the data that can be used for prediction. Once a model has been trained against a representative set of data, it can then be applied to a wider set of data to derive useful information.

Learning can be either supervised or unsupervised. Supervised learning discovers patterns and relationships in the data by comparing the resulting data to a set of known data. Unsupervised learning discovers patterns and relationships in the data without comparing the data to a known answers.

The algorithm vendor determines the specific information that you must provide to use the algorithm. The regression algorithm employs supervised learning. Therefore, the model requires input data and output data.

To use an algorithm, you need to enter its settings, parameters and MaxL DML expressions describing input and output.

Settings are determined by the Data Mining Framework, and as such are the same for all algorithms, but they do influence the operation of the algorithm. For example, the framework provides a setting to define how missing values are treated by the algorithm.

Parameters are specific to each algorithm and determine how the algorithm operates on the data. For example, the clustering algorithm provides a parameter to specify the maximum number of clusters to return.

MaxL DML expressions specify the input and output data for the algorithm. Administration Services Console offers simplified and advanced interfaces for entering these expressions. In the simplified interface, expressions are specified per domain. In the advanced interface, expressions are specified per domain within each accessor. Accessors and domains are determined by the algorithm. In a build task, supervised algorithms such as the Regression algorithm have accessors to define the independent or input data (for example, predictor accessors) and accessors to define the dependent or expected output data (for example, target accessors). Unsupervised algorithms, such as clustering, have a predictor accessor only.

Test and apply tasks generally have input and output accessors.

Accessors consist of domains, which are typically MaxL DML expressions that define components of the accessor. For example, the predictor accessor for the Regression algorithm contains the following domains:

- `Predictor`

  Defines the member or member set combination that determines the predictor domain.

- `Sequence`

  Defines the cases to be traversed for the predictor variable. Sequence is often a subset of the time dimension.

- `External`

  Defines the scope of the predictor (optional).

- `Anchor`

  Defines restrictions from additional dimensions (optional).

The target accessor has the same set of domains as the predictor except that instead of Predictor domain it has Target domain. You write MaxL DML expressions to define the predictor and target accessors. In the simplified interface, you write an expression per domain, regardless of the accessor it comes from.

For example, consider this sample data mining problem:

Given the number of TVs, DVDs, and VCRs sold during a particular period, in the East region, how many cameras were sold in the same period in the East? Restrict sales data to prior year actual sales.

Using the regression algorithm, the predictor and target accessors define the model for this problem. When the values are the same, you can define them in simplified mode as follows:

Domain	Value
`Predictor`	`{[Television], [DVD], [VCR]}`
`Target`	`{[Camera]}`

Domain	Value
`Sequence`	`{[Jan 1].Level.Members}`
`External`	`{[East].Children}`
`Anchor`	`{([2001], [Actual], [Sales])}`

If you need to specify different expressions for similar domains, you should define them in advanced mode; for example:

Domain	**Value**
`Predictor.Predictor`	`{[Television], [DVD], [VCR]}`
`Predictor.Sequence`	`{[Jan 1].Level.Members}`
`Predictor.External`	`{[East].Children}`
`Predictor.Anchor`	`{([2001], [Actual], [Sales])}`
`Target.Target`	`{[Camera]}`
`Target.Sequence`	`{[Jan 1].Level.Members}`
`Target.External`	`{[West].Children}`
`Target.Anchor`	`{([2002], [Actual], [Sales])}`

In both interfaces a predictor component (for example predictor.sequence) and the corresponding target component (target.sequence) must be the same size.

External domain is used when there is a need to build a family of models. For each city in the East ({[East].Children}), the algorithm models camera sales as a function of TV, DVD, and VCR sales. The Data Mining Framework creates, under the same name, a family of results, or models; a separate model or result for each city in the East. If a single model is desired, the external domain may be omitted (left blank).

When you define accessors using the Data Mining Wizard in Advanced mode, you can have mathematical transformations applied to them. For additional information, see "Transformations" on page 593.

➤ To perform a data mining build task, see "Creating or Modifying Build Tasks" in the *Oracle Essbase Administration Services Online Help*.

Training the Model

In the final step of the build task Data Mining Framework executes the algorithm against the data specified by the accessors to build or train the model. During the training process, the algorithm discovers and describes the patterns and relationships in the data that can be used for prediction.

Internally, the algorithm represents the patterns and relationships it has discovered as a set of mathematical coefficients. Later, the trained model can use these patterns and relationships to generate new information from a different, but similarly structured, set of data.

Note:

If you cancel a data mining model while you are training it, the transaction is rolled back.

Testing the Model

After the model is trained, it is ready to use. If you have a known set of existing results, you can test the model against these known results. To test a model, you create a test task. In the test task, you specify a model you have trained and a set of accessors. In addition to accessors specified for input, you specify test accessors that reference the known set of results.

The test task compares the results derived from the trained model to the set of known results you specify. The test task determines if the results match within a specified range of expected error. If the results do not match, you can do any of the following:

- Verify the homogeneity of the known data. If the structure of the known data does not match the structure of the test data, the results will not match.
- Verify the integrity of the known data. Corrupt or incomplete data can cause the test model to fail.
- Verify the integrity of the test input data.
- Consider changing the stringency of the settings. At very least, a less stringent setting that returns a positive test gives you an idea of how closely the trained model compares to known data.

➤ To perform a data mining test task, see "Creating or Modifying Test Tasks" in the *Oracle Essbase Administration Services Online Help*.

Applying the Model

After the model is trained, you can use it on a new set of data. In the apply task you specify a build model you trained and a set of accessors. Applying a model uses the model against a known set of stored data which you specify in the apply task specifications. Generally, the accessor or domain expressions are the same for the predictor domains for a build task and its related apply task. You change the sequence, external, or anchor domain to apply the model to a different set of data. For example, you could change the external domain to specify a different region or country. Or you could use the anchor domain to specify a different year.

In the apply task, the target result data is not known, but is to be predicted by the model.

The apply task applies the model coefficients it generated to the new set of data and generates a set of output data. The Data Mining Framework writes the result data back to the Essbase cube into the locations specified by the target accessor expressions. The apply task generates a named result that you can use to query the result data. For information about viewing the result information, see "Viewing Data Mining Results" on page 590.

➤ To perform a data mining apply task, see "Creating or Modifying Apply Tasks" in the *Oracle Essbase Administration Services Online Help*.

Viewing Data Mining Results

When you execute an apply task, Data Mining Framework writes mining results back to the Essbase cube. Data Mining Framework creates a result record, in XML format, that contains accessors that specify the location of the result data in the cube.

You can view data mining results through the Data Mining node in Administration Services, or using MaxL statements, and an XML-based tool that can access the results.

➤ To view data mining results, see "Managing and Viewing Data Mining Results" in the *Oracle Essbase Administration Services Online Help*.

Scoring the Model

After the model is trained, instead of using an apply task to write back the results to the database you can perform a scoring task to view the results immediately. The input data can come from the cube (Score on Cube) or you can enter data when you execute the task (Score on Data).

For cube scoring you must provide all the expressions required in apply mode with the following exception: the external (if present) and sequence expressions may point to a single position only.

For data scoring you must enter the external expression (if present) for a single position; you must also enter the predictor data as numbers. No sequence expressions are used.

Note:

External expression may appear in apply or scoring tasks only if it was used during build. Otherwise it is omitted.

➤ To perform a data mining scoring task, see "Creating or Modifying Scoring Tasks" in the *Oracle Essbase Administration Services Online Help*.

Accessing Data Mining Functionality

Data Mining Framework is supported by the MaxL and Administration Services interfaces.

MaxL provides a set of statements explicitly for data mining. With MaxL you can perform all data mining functions, including creating, training, testing, scoring, and applying a data mining model.

Sample model templates for each of the algorithms are available through the Administration Services interface. A model template provides an outline of the accessors needed for that algorithm that you can fill in. It also enables you to provide parameters required by the algorithm.

Algorithms

Essbase provides six commonly used algorithms. You can use these algorithms or you can create and register your own algorithms. This topic discusses:

- "Built-in Algorithms" on page 591
- "Creating Algorithms" on page 593

Built-in Algorithms

Essbase supplies the following basic algorithms:

- Regression. Identifies dependencies between a specific value and other values. For example, multilinear regression can determine how the amount of money spent on advertising and payroll affects sales values.
- Clustering. Arranges items into groups with similar patterns. You use the clustering algorithm for unsupervised classification. The algorithm examines data and determines itself how to split the data into groups, or clusters, based on specific properties of the data. The input required to build the model consists of a collection of vectors with numeric coordinates. The algorithm organizes these vectors into clusters based on their proximity to each other. The basic assumption is that the clusters are relatively smaller than the distance between them, and, therefore, can be effectively represented by their respective centers. Hence the model consists of coordinates of center vectors.

  Sequential runs on the same training set may produce slightly different results due to the stochastic nature of the method. You specify the number of clusters to generate, but it is possible the algorithm will find fewer clusters than requested.

  Clusters can provide useful information about market segmentation and can be used with other predictive tools. For example, clusters can determine the kinds of users most likely to respond to an advertising campaign and then target just those users.
- Neural network. Generalizes and learns from data. For example, neural networks can be used to predict financial results.

  You can use the neural net algorithm for both prediction and classification. This algorithm is much more powerful and flexible than linear regression. For example, you can specify multiple targets as well multiple predictors.

  On the other hand, the model generated by the neural net algorithm is not as easy to interpret as that from linear regression.

  One use of neural nets is binary classification. A series of inputs (predictors) produces a set of results (targets) normalized to values between zero and one. For example, a set of behaviors results in values between 0 and 1, with 1 being risky and 0 being risk free. Values in between require interpretation; for example, 0.4 is the high end of safe and 0.6 is the low end of risky.

- Decision tree. Determines simple rules for making decisions. The algorithm results are the answers to a series of yes and no questions. A yes answer leads to one part of the tree and a no answer to another part of the tree. The end result is a yes or no answer. Decision trees are used for classification and prediction. For example, a decision tree can tell you to suggest ice cream to a particular customer because that customer is more likely to buy ice cream with root beer.

  Use the decision tree algorithm to organize a collection of data belonging to several different classes or types. In the build phase, you specify a set of data vectors and provide the class of each. In the apply phase, you provide a set of previously unknown vectors and the algorithm deduces their classes from the model.

  The algorithm constructs a series of simple tests or predicates to create a tree structure. To determine the class of a data vector, the algorithm takes the input data and traverses the tree from the root to the leaves performing a test at each branch.

- Association Rules. Discovers rules in a series of events. The typical application for this algorithm is market basket analysis: people who buy particular items also buy which other items. For example, the result of a market basket analysis might be that men who buy beer also buy diapers.

  You define support and confidence parameters for the algorithm. The algorithm selects sufficiently frequent subsets selected from a predefined set of items. On input it reads a sequence of item sets, and looks for an item set (or its subset), whose frequency in the whole sequence is greater than support level. Such item sets are broken into antecedent-consequent pairs, called rules. Rule confidence is the ratio of its item set frequency to the antecedent frequency in all the item sets. Rules with confidence greater than the given confidence level are added to the list of "confident" rules.

  Although the algorithm uses logical shortcuts during computations, thus avoiding the need to consider all the combinations of the item sets, whose number can be practically infinite, the speed with which the algorithm executes depends on the number of attributes to consider and the frequency with which they occur.

- Naive Bayes. Predicts class membership probabilities. Naive Bayes is a light-weight classification algorithm. It is fast, takes small memory and in a good number of applications behaves quite satisfactory, so you can use it first before going to the decision tree or fully fledged clustering schemes.

  The algorithm treats all the attributes of the case vector as if they were independent of each other. It uses a training sequence of vectors and the theoretical definition of the conditional probability to calculate the probabilities or likelihoods that an attribute with a certain value belongs to a case with a certain class. The model stores these probabilities. In the apply mode the case attributes are used to calculate the likelihood of the case for each class. Then a class with the maximal likelihood is assigned to the case.

For details about the parameters, accessors, and other information used with these algorithms, see the *Oracle Essbase Administration Services Online Help*, which includes a separate topic for each algorithm provided by Essbase.

Creating Algorithms

You can create your own algorithms using Java and register them with Data Mining Framework. In order to be recognized by Data Mining Framework, an algorithm must implement certain interfaces and have a specific signature. These requirements are described in detail in the *Algorithm and Transformation Vendor's Guide* included in the Data Mining Framework SDK.

After a new algorithm is registered, it appears in the list of supplied algorithms and you can use the new algorithm to create build and apply tasks. Data Mining Framework reads the instructions for each parameter in the algorithm from the algorithm signature. The instructions appear in the Build, Score, and Apply Wizard panels where the user sets the algorithm parameters, just like the instructions for the supplied algorithms.

Transformations

Transformations are mathematical operations that you can have applied to accessors when you use the Data Mining Wizard in advanced mode to define a build task model.

Built-In Transformations

Essbase provides the following built-in transformation operations:

- Exponential transformation with a shift: x -> exp (x + s)
- Logarithmic transformation with a shift: log (x + s)
- Power transformation: x -> sign(x)*[x]^p
- Scale transformation: x -> s * x
- Shift transformation: x -> x + s
- Linear transformation: y = b + kx

Creating Transformations

You can create your own transformations using Java and register them with Data Mining Framework. In order to be recognized by Data Mining Framework, a transformation must implement certain interfaces and have a specific signature. These requirements are described in detail in the *Algorithm and Transformation Vendor's Guide* included in the Data Mining Framework SDK. After a new transformation is registered, it appears in the list of supplied transformations that are displayed in the Data Mining Wizard in Advanced mode.

➤ To create and use transformations, see the following topics in the *Oracle Essbase Administration Services Online Help*:

- "Applying Transformations"
- "Managing Data Mining Transformations"

Importing and Exporting Models

Using Predictive Model Markup Language (PMML) format, you can import and export trained data mining models. Importing models enables use of models created outside the Data Mining Framework. You can also use the import-export feature to move models across multiple Essbase Servers.

➤ To import or export data mining models, see “Importing and Exporting Data Mining Models” in the *Oracle Essbase Administration Services Online Help*.

Note:

Extensions added to PMML by other vendors may not be supported. When using an imported model, you must identify an algorithm that most closely matches the algorithm of the imported model.

37 Retrieving Relational Data

In This Chapter

The information in this chapter applies only to block storage databases and is not relevant to aggregate storage databases.

Also see:

- Chapter 57, "Comparison of Aggregate and Block Storage"
- *Oracle Essbase Integration Services System Administrator's Guide*
- *Oracle Essbase Integration Services Online Help*

Integrating Relational Databases with Essbase

Because relational databases can store several terabytes of data, they offer nearly unlimited scalability. Multidimensional databases, generally smaller than relational databases, offer sophisticated analytic capabilities. By integrating a relational database with an Essbase database, you leverage the scalability of the relational database with the conceptual power of the multidimensional database.

By default, when Integration Server creates an Essbase outline, it loads all member levels specified in the metaoutline into a multidimensional database. You can, however, set Integration Server to build to a specified member level (Hybrid Analysis) or build only to the dimension level (Advanced Relational Access). Building down to a specified level produces a smaller multidimensional database and a smaller Essbase outline.

See "Hybrid Analysis" on page 596 and "Advanced Relational Access" on page 602.

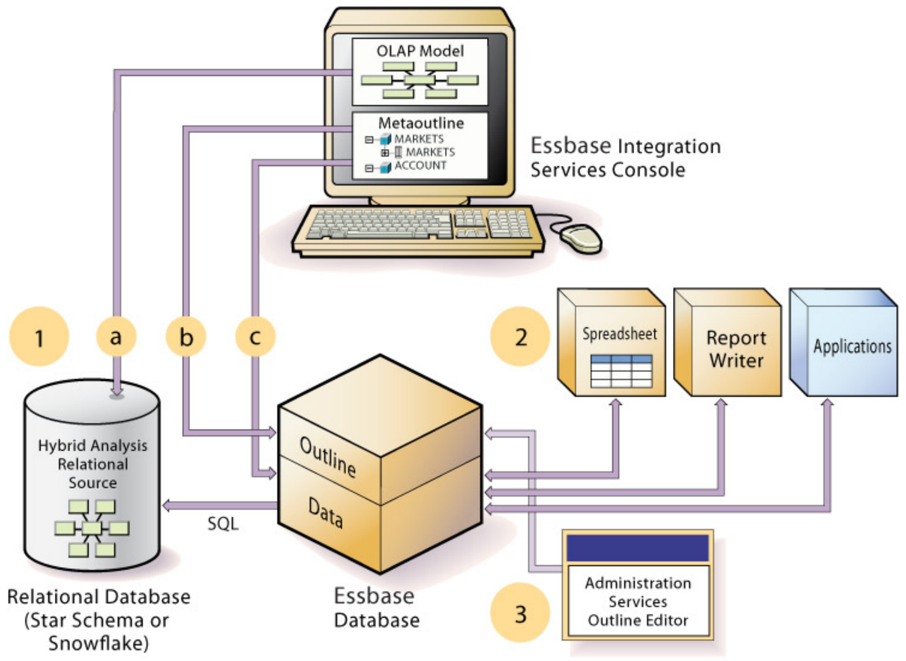
A source relational database can be integrated with an Essbase database by using XOLAP (extended online analytic processing). This a variation on the role of OLAP in business intelligence. Specifically, XOLAP is an Essbase multidimensional database that stores only the outline metadata and retrieves data from a relational database at query time. XOLAP thus integrates 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 best suited to your environment.

See "XOLAP Overview" on page 603.

Hybrid Analysis

Hybrid Analysis eliminates the need to load and store lower-level members and their data within the Essbase database. This feature gives Essbase the ability to operate with almost no practical limitation on outline size and provides for rapid transfer of data between Essbase databases and relational databases. Hybrid Analysis integrates a relational database with an Essbase multidimensional database so that applications and reporting tools can retrieve data directly from both databases. Figure 143 illustrates the Hybrid Analysis architecture:

Figure 143 Hybrid Analysis Architecture


OLAP Model
Metaoutline
MARKETS
MARKETS
ACCOUNT
Essbase Integration Services Console
1
a
b
c
2
Spreadsheet
Report Writer
Applications
Hybrid Analysis Relational Source
SQL
Outline
Data
Relational Database (Star Schema or Snowflake)
Essbase Database
3
Administration Services Outline Editor

Hybrid Analysis Relational Source

The initial step in setting up Hybrid Analysis is to define the relational database as a Hybrid Analysis relational source (**1** in Figure 143).

You define the Hybrid Analysis relational source in Integration Services Console. Through Integration Services Console, you first specify the relational data source for the *OLAP model*. The OLAP model is a schema that you create from tables and columns in the relational database. To build the model, Integration Services accesses the star schema of the relational database (**a** in Figure 143).

Using the model, you define hierarchies and tag levels whose members are to be enabled for Hybrid Analysis. You then build the *metaoutline*, a template containing the structure and rules for creating the Essbase outline, down to the desired Hybrid Analysis level. The information enabling Hybrid Analysis is stored in the OLAP Metadata Catalog, which describes the nature, source, location, and type of data in the Hybrid Analysis relational source.

Next, you perform a *member load*, which adds dimensions and members to the Essbase outline (**b** in Figure 143). When the member load is complete, you run a data load to populate the Essbase database with data (**c** in Figure 143). At this point, the Hybrid Analysis architecture is in place:

- The lower-level members and their associated data remain in the relational database.
- The data in the relational database is mapped to the Essbase outline that is defined by Hybrid Analysis.
- The outline resides in the Essbase database.

  Metadata is data that describes values within a database. The metadata that defines the Hybrid Analysis data resides in the Essbase outline and in the Integration Services metaoutline on which the Essbase outline is based. Any changes that are made to Hybrid Analysis data in an OLAP model or metaoutline that is associated with an Essbase outline must be updated in the outline to ensure accuracy of the data reported in Essbase. See "Managing Data Consistency in Hybrid Analysis" on page 601.
- Upper-level members and their associated data reside in the Essbase database.

Data Retrieval

Applications and reporting tools, such as spreadsheets and Report Writer interfaces, can retrieve data directly from both databases (**2** in Figure 143). Using the dimension and member structure defined in the outline, Essbase determines the location of a member and then retrieves data from either the Essbase database or the Hybrid Analysis relational source. If the data resides in the Hybrid Analysis relational source, Essbase retrieves it through SQL commands. See "Retrieving Hybrid Analysis Data" on page 597.

To modify the outline, you can use Outline Editor in Administration Services to enable or disable dimensions for Hybrid Analysis on an as-needed basis. (**3** in Figure 143). See "Using Outline Editor with Hybrid Analysis" on page 600 and the *Oracle Essbase Integration Services System Administrator's Guide*.

Retrieving Hybrid Analysis Data

For information on defining a Hybrid Analysis relational source in Integration Services Console, see the *Oracle Essbase Integration Services Online Help*.

In Hybrid Analysis, applications and reporting tools can retrieve data directly from relational and Essbase databases by using the following tools:

- Spreadsheet Add-in
- Smart View

- Report Writer
- Oracle's Hyperion® Web Analysis
- Third-party applications

Note:

The Essbase database and the relational database must be registered to the same ODBC data sources, and Integration Services must use the same source name for both databases.

Because data is being accessed from the Hybrid Analysis relational source and the Essbase database when you perform calculations or generate reports, data retrieval time may increase with Hybrid Analysis; however, most capabilities of Essbase data retrieval operations are available with Hybrid Analysis, including pivot, drill-through, and other metadata-based methods.

Defining the Hybrid Analysis Retrieval Environment

Use the configuration settings listed in Table 60 to enable and define Hybrid Analysis for an Essbase Server or specific applications and databases:

Table 60 Hybrid Analysis Retrieval Configuration Settings

Setting	Description
HAENABLE	Enables retrieval of members from a Hybrid Analysis relational source
HAMAXNUMCONNECTION	Sets the maximum number of connections per database that Essbase can keep active against the relational database
HASOURCEDSNOS390	Enables access to DB2 data sources on an OS/390 system
HAMAXNUMSQLQUERY	Sets the maximum number of SQL queries that can be issued against relational database fact tables per Essbase query session
HAMAXQUERYTIME	Sets the maximum time limit per query for SQL queries from a Hybrid Analysis Relational Source
HAMAXQUERYROWS	Sets the maximum number of rows that can be returned per SQL query issued on behalf of an Essbase query
HARETRIEVENUMROW	Sets the maximum number of rows resulting from an SQL query to process at one time
HARAGGEDHIERARCHY	Enables support of null values in columns of dimension tables that are used to create dimensions for Hybrid Analysis-enabled outlines
HAMEMORYCACHESIZE	Sets the amount of memory reserved to cache queried members from a Hybrid Analysis relational source

See the *Oracle Essbase Technical Reference*.

Retrieving Hybrid Analysis Data with Spreadsheet Add-in

Use the Enable Hybrid Analysis option in the Essbase Options dialog box in Spreadsheet Add-in to drill down to members in the Hybrid Analysis relational source.

See *Oracle Essbase Spreadsheet Add-in User's Guide*, *Oracle Essbase Spreadsheet Add-in Online Help*.

Hybrid Analysis supports the following options in Spreadsheet Add-in:

- Drill-down:
  - Next Level (children)
  - All Levels (all descendants)
  - Bottom Level (level 0)
  - All Siblings (all members with common parent)

  For best performance, use children, bottom-level, or siblings zoom-ins; avoid descendents zoom-ins.
- Drill-up: Parent drill-up

  The drill-up on a relational member always takes you to the leaf-level member in the Essbase outline, and not to the immediate parent of the relational member.

Retrieving Hybrid Analysis Data with Smart View

Hybrid Analysis is automatically enabled in Smart View.

Hybrid Analysis supports the Parent drill-up option in Smart View. The drill-up on a relational member always takes you to the leaf-level member in the Essbase outline and not to the immediate parent of the relational member.

See the *Oracle Hyperion Smart View for Office Online Help*.

Retrieving Hybrid Analysis Data with Report Writer

In Report Writer, commands enable and disableHybrid Analysis:

- <HYBRIDANALYSISON enables a report script to retrieve the members of a dimension that is enabled for Hybrid Analysis.
- <HYBRIDANALYSISOFF prevents a report script from retrieving the members of a dimension that is enabled for Hybrid Analysis.

The <ASYM and <SYM commands are not supported with Hybrid Analysis. If they are present in a report, errors may result. The <SPARSE command is ignored in reports retrieving data from a Hybrid Analysis relational source and does not generate errors.

The following is a sample Report Writer script that uses the IDESCENDANTS command to return Hybrid Analysis data:

```
<PAGE (Accounts, Scenario, Market)
Sales
Actual
```

```
<Column (Time)
<CHILDREN Time
<Row (Product)
<IDESCENDANTS 100-10
!
```

Retrieving Hybrid Analysis Data with Web Analysis

When you use Web Analysis, the procedures for retrieving Hybrid Analysis data are the same as those for retrieving data that is not defined for Hybrid Analysis. (See the Web Analysis documentation.)

For optimal performance when retrieving Hybrid Analysis data with Web Analysis, consider the following guidelines:

- Place dimensions that are enabled for Hybrid Analysis along the rows when constructing queries in Web Analysis.
- Avoid using sort options.
- Use the children operator when drilling down to a hierarchy; do not use the descendants operator.
- Additional processing in the form of restrictions or Top/Bottom retrievals may cause slower query times.

Using Outline Editor with Hybrid Analysis

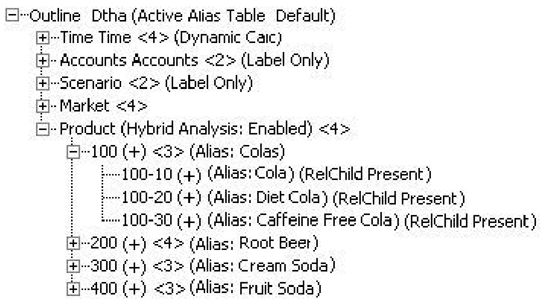
In Outline Editor, you can toggle the Hybrid Analysis option button to enable or disable Hybrid Analysis for each dimension that is defined for Hybrid Analysis in Integration Services Console. If you open an outline that is not defined for Hybrid Analysis, the Hybrid Analysis option button is not displayed on the toolbar.

Note:

When Hybrid Analysis is disabled for a dimension, the user is unable to see and drill through to the Hybrid Analysis data associated with the dimension; however, the members of the dimension remain visible in Outline Editor.

Figure 144 is an example of how an outline defined for Hybrid Analysis appears in Outline Editor. Note that dimensions that are enabled for Hybrid Analysis are identified to distinguish them from dimensions that are not enabled for Hybrid Analysis.

Figure 144   Example of Hybrid Analysis in Outline Editor



```
Outline  Dtha (Active Alias Table  Default)
  Time Time <4> (Dynamic Calc)
  Accounts Accounts <2> (Label Only)
  Scenario <2> (Label Only)
  Market <4>
  Product (Hybrid Analysis: Enabled) <4>
    100 (+) <3> (Alias: Colas)
      100-10 (+) (Alias: Cola) (RelChild Present)
      100-20 (+) (Alias: Diet Cola) (RelChild Present)
      100-30 (+) (Alias: Caffeine Free Cola) (RelChild Present)
    200 (+) <4> (Alias: Root Beer)
    300 (+) <3> (Alias: Cream Soda)
    400 (+) <3> (Alias: Fruit Soda)
```

Managing Data Consistency in Hybrid Analysis

When you create a Hybrid Analysis relational source, the data and metadata are stored and managed in the relational database and in the Essbase database:

- Lower-level members and their associated data remain in the relational database.
- Data in the relational database is mapped to the Essbase outline that is defined for Hybrid Analysis.
- The outline resides in the Essbase database.
- Upper-level members and their associated data reside in the Essbase database.

Because data and metadata exist in different locations, information may become out of sync.

Essbase depends upon the OLAP Metadata Catalog in Integration Services to access the Hybrid Analysis relational source. At Essbase database startup time, Essbase Server checks the number of dimensions and members of the Essbase outline against the related metaoutline.

Changes made to the associated OLAP model or metaoutline during an Integration Services session are not detected by the Essbase Server until the Essbase database is started again. Undetected changes can cause data inconsistency between the Essbase database and the Hybrid Analysis relational source.

If changes are made in the Hybrid Analysis relational source, and members are added or deleted in the OLAP model or metaoutline, such changes can cause the Essbase outline to be out of sync with the metaoutline on which it is based. These types of changes and their effect on the hierarchical structure of a dimension are not reflected in the Essbase database until the outline build and data load process is completed through Integration Services Console.

In Administration Services, the Restructure Database dialog box has a check box that enables a warning whenever a restructuring affects an outline containing a Hybrid Analysis relational source. Such a problem occurs, for example, if members with relational children are moved or deleted.

Warnings are listed in the application log. You should decide whether the warnings reflect a threat to data consistency. To view the application log, see "Viewing the Essbase Server and Application Logs" on page 742.

The Essbase administrator has the responsibility to ensure that the Essbase multidimensional database, the relational database, and the Integration Services OLAP model and metaoutline remain in sync. Administration Services and Integration Services Console provide commands that enable the administrator to perform consistency checks and make the appropriate updates.

See Chapter 48, “Ensuring Data Integrity” and Chapter 53, “Optimizing Database Restructuring.”

Managing Security in Hybrid Analysis

The Essbase administrator determines access to the Hybrid Analysis relational source on an individual Essbase user level. Access for Hybrid Analysis is governed by the same factors that affect overall Essbase security:

- Permissions and access levels for individual users and groups of users
- Permissions and access levels for the server, application, or database
- Specific database access levels for particular database members

If a security filter enables you to view only the relational children of the level 0 members that you have access to in Essbase, then you cannot view the relational children of the level 0 members that you do not have access to in Essbase.

Assume that you have the following outline of the Market dimension, where San Francisco and San Jose are relational children of California, and Miami and Orlando are relational children of Florida:

```
Market
  West
    California
      San Francisco
      San Jose
  East
    Florida
      Miami
      Orlando
```

In this example, if a filter allows you to view only level 0 member California and its descendants, you can view California and its relational children, San Francisco and San Jose; however, you cannot view the children of level 0 member Florida.

See the following chapters:

- Chapter 38, “User Management and Security”
- Chapter 39, “Controlling Access to Database Cells”
- Chapter 40, “Security Examples in Native Security Mode”

Advanced Relational Access

Integration Services uses Advanced Relational Access to give Essbase users direct access to data from relational databases or data warehouses. In Integration Services Console, Advanced Relational Storage is enabled at the metaoutline level. When the Relational Storage option is selected, all members of all nonaccounts dimensions are automatically enabled for relational storage. Alternatively, you can enable relational storage on selected nonaccounts dimensions.

When a metaoutline is enabled for Advanced Relational Access, users are able to query directly on relationally stored members. Queries to the database are made using multidimensional expressions (MDX), which are translated into SQL statements. Dimension members are accessed directly from the relational data source.

Note:

For information on enabling Advanced Relational Access, see the *Oracle Essbase Integration Services Online Help* and *Oracle Essbase Integration Services System Administrator's Guide*.

SAP R/3

Essbase provides an SAP R/3 adapter that helps SAP experts identify and integrate SAP R/3 data into Essbase and other Oracle Hyperion application products. When enabled, an SAP R/3 node is available in the Administration Services Console. After setting up the feature, you can open an SAP instance under the node and provide information from which Administration Services defines a logical schema that can be used by Integration Services to build an Essbase outline.

Working with SAP R/3 includes the following process:

1. Install the SAP R/3 adapter (a onetime requirement).

   See the *Oracle Hyperion Enterprise Performance Management System Installation and Configuration Guide*.
2. In Administration Services Console, per specific SAP R/3 database:
   - Configure the data source (see “Connecting to SAP Data Sources in the *Oracle Essbase Administration Services Online Help*).
   - Open the corresponding instance under the SAP R/3 node and log in.
   - Create logical tables and store logical table definitions into the relational database as logical schema.
   - Define the logical table joins, define filter criteria, save and populate logical tables from SAP tables, saving them as a logical schema.
3. In Integration Services, from the tables in the saved logical schema, create an OLAP model and metaoutline, and perform a member load.

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 best suited to your environment.

OLAP and XOLAP store the metadata outline and the underlying data in different locations:

- In OLAP, the metadata is located in the Essbase database, and the underlying data is also located in the Essbase database.
- In XOLAP, the metadata is located in the Essbase database while 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 because these differences affect the functionality of OLAP and XOLAP.

OLAP lends itself to traditional relational data storage and data analysis. XOLAP lends itself to operations supported in mixed or "hybrid" environments such as Hybrid Analysis and Advanced Relational Access (familiar to users of Essbase and Integration Services). Many of the basic concepts of Hybrid Analysis and Advanced Relational Access have been folded into the functionality of XOLAP cubes in Oracle Essbase Studio.

For information on guidelines and restrictions in using XOLAP and how to designate models for XOLAP, see the Essbase Studio online help.

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 in Essbase Studio.
2. The cube is deployed in Essbase Studio; 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 which can access an Essbase database.
4. Essbase dynamically generates the required SQL to retrieve the data from the source relational database.

Guidelines for Using XOLAP

XOLAP has several restrictions. There are also several usages not supported in XOLAP.

Restrictions For XOLAP

XOLAP has the following restrictions:

- No editing of an XOLAP cube is allowed. If you wish to modify an outline, you must, instead, create a new outline in Oracle Essbase Studio. XOLAP operations will not automatically incorporate any changes in the structures and the contents of the dimension tables after an outline is created.
- When derived text measures are used in cube schemas to build an Essbase model, XOLAP is not available for the model.
- XOLAP can be used only with aggregate storage. The database is automatically duplicate-member enabled.

- 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.

Usages Not Supported in XOLAP

XOLAP does not support use of the following

- Flat files
- Ragged hierarchies
- Alternate hierarchies
- Recursive hierarchies
- Calendar hierarchies
- Filters
- Typed measures
- User defined members at the leaf level
- Multiple relational data sources

Part VI

Designing and Managing a Security System

In Designing and Managing a Security System:

38 User Management and Security

In This Chapter

The information in this chapter applies to block storage and aggregate storage databases.

Also see the *Oracle Hyperion Enterprise Performance Management System Security Administration Guide*.

About Using Shared Services with Essbase

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

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

By default, after installation, Essbase Administration Server and Essbase Server are in native security mode. You can continue to use Essbase in native security mode to manage security for

Essbase applications, databases, and artifacts. There is no change in behavior for Essbase in native security mode, except when using native security mode with external authentication. See "Continuing to Use Essbase in Native Security Mode" on page 623.

To use Shared Services security, you must migrate any Essbase Server applications and any existing Essbase users and groups to Shared Services. See "Migrating Essbase to Shared Services" on page 619.

Essbase User Roles for Shared Services

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

- Product-specific roles

  Examples of Essbase roles are Administrator and Database Manager. All Essbase roles are specific to a Shared Services application (the permissions granted to the user by the role apply only to the specific application for which the role is assigned, and not to all applications).
- 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). An exception is the Provisioning Manager role, which is specific to an application. For information on Shared Services roles, see the *Oracle Hyperion Enterprise Performance Management System Security Administration Guide*.

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

You can provision a user with the following roles for an Essbase Server:

- Administrator
- Create/Delete Application
- Server Access

You can provision a user with the following roles for an application:

- Application Manager
- Database Manager
- Calc
- Write
- Read
- Filter
- Start/Stop Application

In Shared Services Console, roles belonging to Essbase are grouped under the Essbase node; roles belonging to Essbase applications are grouped under the application nodes.

Note:

There is no concept of provisioning an Administration Services Administrator user role through Shared Services. When migrated, an Administration Services Administrator is assigned no roles in Shared Services.

Table 61 lists the user roles that are specific to Essbase and the Shared Services role of Provisioning Manager, which is application-specific. The table shows the corresponding tasks that each user can perform.

Table 61 Essbase and Shared Services User Roles and Tasks

User Role	Task Description
Project Manager	(A Shared Services role) Creates and manages projects within Shared Services.
Administrator (previously Supervisor)	Full access to administer the server, applications, and databases. **Note:** The Provisioning Manager role, which is a Shared Services application-specific role, is automatically assigned when you migrate Essbase Administrators (previously known as Supervisors). However, when you create an Essbase Administrator in Shared Services Console, you must manually assign the Provisioning Manager role. Users with the Provisioning Manager role can provision users and groups with roles for applications
Create/Delete Application	Ability to create and delete applications and databases within applications. Includes Application Manager and Database Manager permissions for the applications and databases created by this user.
Server Access	Ability to access any application or database that has a minimum access permission other than none. **Note:** When you assign security at the Essbase application level, you must also assign the user the Server Access role for the Essbase Server that contains the application (unless the user already has another Essbase Server level role, for example Create/Delete Application).
Application Manager (previously Application Designer)	Ability to create, delete, and modify databases and application settings within the particular application. Includes Database Manager permissions for databases within the application. **Note:** The Provisioning Manager role is automatically assigned when you migrate Essbase Application Managers. However, when you create an Essbase Application Manager in Shared Services Console, you must manually assign the Provisioning Manager role.
Database Manager (previously Database Designer)	Ability to manage databases (for example, to change the database properties or cache settings), database artifacts, locks, and sessions within the assigned application.

User Role	Task Description
Calc	Ability to calculate, update, and read data values based on the assigned scope, using any assigned calculations and filter.
Write	Ability to update and read data values based on the assigned scope, using any assigned filter.
Read	Ability to read data values.
Filter	Ability to access specific data and metadata according to the restrictions of a filter.
Start/Stop Application	Ability to start and stop an application or database.

Essbase Projects, Applications, and Databases in Shared Services

Shared Services artifacts include projects, applications, user roles, users, and groups. When you assign access to a user or group in Shared Services, you *provision* the user or group with a *role* for an *application*. See the *Oracle Hyperion Enterprise Performance Management System Security Administration Guide*.

Shared Services and Essbase both use the term "application." Essbase uses "application" to refer to a container for databases. Shared Services uses "application" to refer to an artifact for which you provision users. In this document, "application" refers to a Shared Services application, unless an Essbase application is specifically stated. In most cases, an Essbase application maps to a Shared Services application, so the distinction is unnecessary.

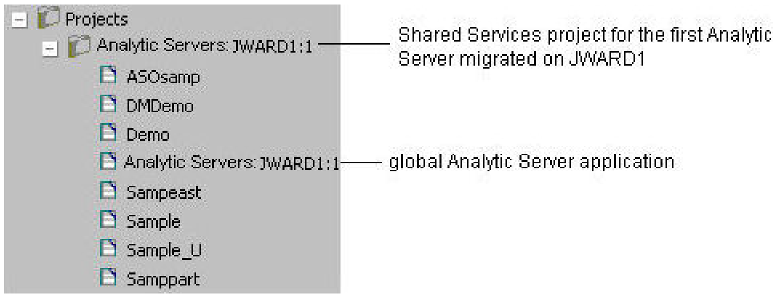
For Essbase, migration is done at the Essbase Server level. When you migrate an Essbase Server to Shared Services, a Shared Services project is created for the Essbase Server. The project is named `Essbase Servers:`*`machineName`*`:` *`EssbaseServer#`* where *`machineName`* is the Essbase Server computer name and *`EssbaseServer#`* is the sequence number. If you migrate multiple Essbase Servers on the same computer, each Essbase Server migrated gets a different sequence number (*`EssbaseServer#`*). Also, if you delete the security file and remigrate an Essbase Server, each successful migration creates a new server project with a new sequence number. You can delete unwanted projects in Shared Services Console.

Essbase automatically creates the following applications within the project and automatically registers the applications with Shared Services:

- An application named `Essbase Servers:`*`machineName`*`:`*`EssbaseServer#`*, which is the same name as the Shared Services project. This application, which allows you to specify security at the Essbase Server level, is known as the *global Essbase Server application*. After migration, you can rename the Shared Services project; however, the global Essbase Server application name is not renamed.
- A Shared Services application for each Essbase application on the Essbase Server. In Shared Services, if an Essbase application contains multiple databases, the databases must have the same user security access levels. (However, users can have different calculation script and

database filters assigned for databases within the same application. See "Assigning Database Calculation and Filter Access" on page 616).

Figure 145 Shared Services Essbase Server project (Essbase Servers: JWARD1:1) and global Essbase Server application (Essbase Servers: JWARD1:1).


Projects
Analytic Servers: JWARD1:1
ASOsamp
DMDemo
Demo
Analytic Servers: JWARD1:1
Sampeast
Sample
Sample_U
Samppart
Shared Services project for the first Analytic Server migrated on JWARD1
global Analytic Server application

Once you have migrated to Shared Services, when you create an application and database in Essbase, a corresponding Shared Services application is created within the Essbase Server project, and the application is automatically registered with Shared Services.

Essbase Users and Groups in Shared Services

When you migrate to Shared Services, all native Essbase users and groups that do not already exist in an external authentication directory are converted to native Shared Services users and groups in the native Shared Services user directory and are given equivalent roles. Externally authenticated users are registered with Shared Services but are still stored in their original authentication directory. See "User and Group Migration" on page 621.

After you have migrated to Shared Services, you must create and manage users and groups in Shared Services Console, or through the external user directory. See the *Oracle Hyperion Enterprise Performance Management System Security Administration Guide*.

When users and groups are stored in an external authentication directory from any supported authentication provider, you must be sure that identical names for users and groups are not used, even if the identically-named users or groups reside in different directories from the same authentication provider (for example, different directories from the same LDAP-based authentication provider) or in directories from different authentication providers (for example, an LDAP-based directory and an MSAD directory).

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 Administrator role, any subgroups (and users in the groups) inherit the Essbase Administrator role.

Note:

In Essbase, when you copy an application or database and the target Essbase Server is in Shared Services security mode, user and group security is not copied with the application. Use the copy provisioning functionality in Shared Services Console to copy security for an application.

Assigning Access to Users in Shared Services

Shared Services Console provides a centralized UI where you can perform user management tasks for Oracle Hyperion products. The Shared Services Console launches Essbase screens, which allow you to assign security access to database filters and calculation scripts. In Shared Services security mode, you use Shared Services Console, MaxL, or the API to manage security. (Some restrictions exist when managing security using MaxL or the API. See the *Oracle Essbase Technical Reference* and the *Oracle Essbase API Reference*.) In Administration Services Console you can only view security information.

For information on assigning access to users and groups and viewing a report of users, groups, and provisioned roles for each application, see the *Oracle Hyperion Enterprise Performance Management System Security Administration Guide*.

Launching and Logging In to Shared Services Console

To manage Essbase users in Shared Services Console, you must log in to the console as a user who is provisioned with the following Shared Services roles:

- Provisioning Manager role for the appropriate Essbase Server or applications
- Directory Manager role for the appropriate authentication directory

When you launch Shared Services Console from Administration Services, you automatically log in to Shared Services Console as the Essbase user that connects the Essbase Server that you are accessing.

Note:

In Shared Services security mode, you must use the same user to log in to Administration Services Console as you use to connect the Essbase Server.

When you launch Shared Services Console from a browser, you log in as whichever user is appropriate. For example, you must log in as a Shared Services Administrator to provision an Essbase Administrator with the Directory Manager role, so that he or she can create and delete users.

➤ To launch Shared Services Console, see "Launching Shared Services Console" in the *Oracle Essbase Administration Services Online Help*.

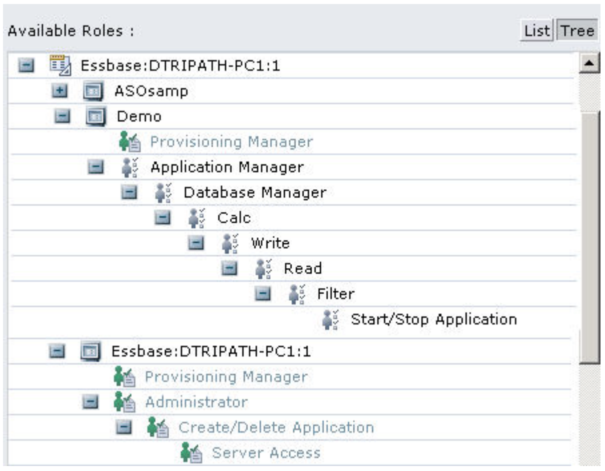
Assigning Server Access

To specify security at the Essbase Server level in Shared Services security mode (for example, provisioning a user with the Provisioning Manager role for all Essbase applications on an Essbase Server), provision the user with the appropriate role for the global Essbase Server application; that is, the Shared Services application that represents the Essbase Server. See "Essbase Projects, Applications, and Databases in Shared Services" on page 612.

Note:

When you provision a user with the Essbase Administrator role, you must also manually provision the user with the Provisioning Manager role for the Essbase Server and for each Essbase application on the server. (When you migrate an Essbase Administrator, the Provisioning Manager role is automatically assigned.)

Figure 146 Shared Services Console provisioning panel, displaying the roles available for the Essbase Server DTRIPATH-PC1 and the Demo application



```
Available Roles :                                   List | Tree
Essbase:DTRIPATH-PC1:1
  ASOsamp
  Demo
    Provisioning Manager
    Application Manager
      Database Manager
        Calc
          Write
            Read
              Filter
                Start/Stop Application
  Essbase:DTRIPATH-PC1:1
    Provisioning Manager
    Administrator
      Create/Delete Application
        Server Access
```

Assigning Application Access

To specify security at the Essbase application level in Shared Services security mode (for example, provisioning a user with the Database Manager role for the Sample application) provision the user with the appropriate role for the application.

Note:

When you assign security at the Essbase application level, you must also assign the user the Server Access role for the Essbase Server that contains the application (unless the user already has another Essbase Server level role, for example Create/Delete Application).When you provision a user with the Application Manager role, you must also manually provision the user with the Provisioning Manager role for the appropriate application. (When you migrate an Essbase Application Manager, the Provisioning Manager role is automatically assigned).

You can set minimum permissions for an application, for example, if you want all users to have at least write access to all databases in an application. The default setting is None, meaning that no minimum permission is set; all users can access the application according to their roles.

➤ To set the minimum permission for an application, see "Setting Minimum Permissions for Applications" in the *Oracle Essbase Administration Services Online Help*.

Assigning Database Calculation and Filter Access

After provisioning users for Essbase applications in Shared Services Console, you can assign more granular access permissions to users and groups for a specific Essbase application and database. For example, after assigning a user access to an application and assigning the user's role for the application, you can assign an Essbase filter to the user, or assign the user access to a specific calculation script.

When you select an Essbase application from Shared Services Console, a screen is displayed, listing the users and groups provisioned to that application. On this screen, you select the users and groups to which you want to assign additional permissions. After clicking Next, select the database you want to work with, and then use the drop-down lists to assign filter and calculation script access to selected users and groups. For descriptive information about these two screens, click the Help button on one of these screens to display a context-sensitive help topic.

➤ To specify access permissions in Shared Services, use a tool:

Tool	Topic	Location
Administration Services	Assigning Database Calculation and Filter Access	*Oracle Essbase Administration Services Online Help*
MaxL	**grant**	*Oracle Essbase Technical Reference*

When you assign database calculation and filter access, you automatically log in to Administration Services and Essbase as the Shared Services Console logged-in user. This user must be an Essbase Administrator, Application Manager, or Database Manager. The user must have the Provisioning Manager role for the appropriate application(s).

You cannot assign database calculation or filter access to an Essbase Administrator or Application Manager.

➤ To refresh Essbase with database calculation and filter access security information for newly provisioned users, click **Refresh**.

Although you can assign access to database filters and calculation scripts through Shared Services Console, you must create the filters and calculation scripts in Essbase. For information on creating database filters, see Chapter 39, "Controlling Access to Database Cells."

Assigning Application Access Type

Essbase and Planning have the concept of an "application access type" for Essbase and Planning users. For example, when an Essbase user is created using any Essbase administration tool, the user is automatically assigned the application access type "Essbase"; when a Planning user is created using the Planning interface, the user is automatically assigned the application access type “Planning.” A user’s application access type specifies whether the user has access to Essbase applications only, to Planning applications only, or to both.

When you select a global Essbase Server application from Shared Services Console, a screen is displayed, listing the users and groups provisioned to that application. On this screen, you select the users and groups for which you want to assign application access type. After clicking Next, use the drop-down list to assign application access type to the selected users and groups. For descriptive information about these two screens, click the Help button on one of these screens to display a context-sensitive help topic.

➤ To specify application access types for users in Shared Services, use a tool:

Tool	Topic	Location
Administration Services	Assigning Application Access Type for Users in Shared Services	*Oracle Essbase Administration Services Online Help*
MaxL	**create user**	*Oracle Essbase Technical Reference*

When you assign database calculation and filter access, you automatically log in to Administration Services and Essbase as the Shared Services Console logged-in user. This user must be a valid Essbase Administrator and must have the Provisioning Manager role for the appropriate applications.

➤ To refresh Essbase with application access type information for newly provisioned users, click **Refresh.**

Synchronizing Security Information Between Shared Services and Essbase

This topic provides information on synchronizing Essbase security with Shared Services security. (When the security information is out of sync, the user, group, and application information displayed in Essbase may be different from that in Shared Services.)

User synchronization refers to the process of ensuring that Essbase reflects the latest security information for a specific user and any related groups.

Refresh refers to the process of ensuring that Essbase reflects the latest security information for all users, groups, and applications on an Essbase Server.

Using the Essbase configuration settings CSSSYNCLEVEL and CSSREFRESHLEVEL in the `essbase.cfg` file, you can set user synchronization and refresh to happen in the following ways:

- Automatically at sync points.
- When an Administrator requests a refresh of the security information.
- When a user selects a database (user synchronization only).

User Synchronization Using CSSSYNCLEVEL

The CSSSYNCLEVEL configuration setting controls how Shared Services synchronizes security information for a specific user and any related groups when the user logs in to Essbase or selects a database.

- `CSSSYNCLEVEL AUTO`: Shared Services synchronizes security information for a specific user and any related groups when the user logs in to Essbase or selects a database.

  User e-mail ID and description are not synchronized at user login or when selecting a database. E-mail ID and description are synchronized only following a requested or periodic (using the SHAREDSERVICESREFRESHINTERVAL configuration setting in `essbase.cfg`) full refresh of security information.

- `CSSSYNCLEVEL NONE`: User information is not synchronized when a user logs in to Essbase or selects a database. Shared Services synchronizes user information only when an Administrator, Application Manager, or Database Manager requests a refresh of security information.

  If NONE is specified, when you provision a user with an Essbase Server role, you must request a refresh of security information to enable the user to log in.

- `CSSSYNCLEVEL DBSELECT`: User information is synchronized when a user selects a database but not when the user logs in to Essbase.

  If DBSELECT is specified, when you provision a user with an Essbase Server role, you must request a refresh of security information to enable the user to log in.

User Refresh Using CSSREFRESHLEVEL

The CSSREFRESHLEVEL configuration setting controls how Shared Services refreshes the status of all users, groups, and applications for an Essbase Server at Essbase Server startup.

- `CSSREFRESHLEVEL AUTO`: Shared Services automatically refreshes the status of all users, groups, and applications for an Essbase Server at Essbase Server startup.
- `CSSREFRESHLEVEL MANUAL`: Shared Services refreshes security information only when an Administrator requests a refresh of security information.

➤ To request a refresh of security information, use a tool:

Tool	Topic	Location
Administration Services	Refreshing Security From Shared Services	*Oracle Essbase Administration Services Online Help*
MaxL	**alter system**	*Oracle Essbase Technical Reference*

Note:

The information in this topic does not apply to changes made to access permissions for database filters and calculation scripts, which are synchronized immediately.

Role Requirements For Refreshing Security

You must have the following roles to refresh security:

- Refresh security for the Essbase Server: Essbase Administrator.
- Refresh security for an application: Administrator, Application Manager, or Database Manager.
- Refresh security for a user: Users can synchronize their own security. An Essbase Administrator can synchronize security for all users.

Scheduling Security Refreshes

You can specify periodic, automatic refreshes of Essbase security information from Shared Services. For example, you can specify that Essbase refresh security information from Shared Services every 60 minutes.

➤ To schedule information refreshes from Shared Services, see the SHAREDSERVICESREFRESHINTERVAL configuration setting in the *Oracle Essbase Technical Reference*.

Note:

The CSSREFRESHLEVEL setting does not affect the SHAREDSERVICESREFRESHINTERVAL setting.

Migrating Essbase to Shared Services

After installation, Essbase and Administration Services are in native security mode. To use Shared Services, you must migrate to Shared Services security mode. For Essbase, migration is done at the Essbase Server level. Once you have converted to Shared Services security mode, you cannot convert back to native security mode.

Essbase Administration Server can run in Shared Services security mode with Essbase Server running in native security mode. However, if any Essbase Server that you administer from Administration Services Console runs in Shared Services security mode, Essbase Administration Server must also.

➤ To migrate Essbase Server, Essbase Administration Server, and users and groups to Shared Services, use a tool:

Tool	Topic	Location
Administration Services	Converting Essbase Server and Migrating Users to Shared Services	*Oracle Essbase Administration Services Online Help*
MaxL	**alter system**	*Oracle Essbase Technical Reference*

You must be an Essbase Administrator to run a migration.

For Essbase Administration Server, if you ran Oracle's Hyperion Enterprise Performance Management System Configurator after installation and specified a Shared Services server and login, at that point, Essbase Administration Server is converted to Shared Services security mode. You can view the Shared Services configuration information in the Essbase Administration Server properties window (Configuration tab). You can then choose to migrate Administration Services users to Shared Services.

➤ To migrate Administration Services users or to remigrate any Essbase users and groups that failed migration, use a tool:

Tool	Topic	Location
Administration Services	Migrating Users to Shared Services	*Oracle Essbase Administration Services Online Help*
MaxL	**display user** **display group** **alter user** **alter group**	*Oracle Essbase Technical Reference*

You must be an Administration Services Administrator to migrate Administration Services users.

Note:

Before migrating users, groups, and applications to Shared Services, ensure that the NETDELAY and NETRETRYCOUNT configuration settings are high enough to allow the migration to complete. Set NETDELAY to at least 3 hours, possibly more, depending on the size of the security file. Return the settings to their original values once the migration is complete. Specify these settings in the client `essbase.cfg` file, which you place in the *`ESSBASEPATH`*`/bin` folder of the client computer from which you launch the migration. For example, if you use Administration Services Console to launch the migration, the client `essbase.cfg` file must be in the *`ESSBASEPATH`*`/bin` folder on the computer on which Essbase Administration Server is installed.

Essbase automatically creates a backup of the security file before and after migration (`essbase.bak_preUPM` and `essbase.bak_postUPM`). Oracle suggests that you manually back up these files to a safe location.

The Administration Services Essbase Server Properties window displays information on whether the server is in Shared Services security mode.

Application and Database Migration

After you have migrated to Shared Services, a project is created for each Essbase Server that you migrate. The project contains a Shared Services application for each Essbase application on the migrated server. See "Essbase Projects, Applications, and Databases in Shared Services" on page 612.

User and Group Migration

When you migrate to Shared Services, all native Essbase users and groups that do not already exist in an external authentication directory are converted to native Shared Services users and groups in the native Shared Services user directory and are given equivalent roles. For example, a native Essbase Administrator (previously known as Supervisor) becomes a Shared Services user with the Essbase Administrator and the Provisioning Manager roles assigned, and a native Essbase user with Calc privileges on a database becomes a Shared Services user with the Calc role assigned on the application that contains the database. During migration, Administrators and Application Managers are automatically given the Provisioning Manager role for the appropriate applications.

Note:

When Essbase runs in Shared Services mode, the Essbase create/delete user privilege becomes obsolete. You must be an Essbase administrator to create/delete Essbase users, and you must additionally be a Shared Services administrator to create/delete users in Shared Services.

Any externally authenticated users are registered with Shared Services but remain stored in their original authentication directory. If a user directory is not running, the entire migration fails.

Users created using custom authentication are not migrated unless a matching user is already in Shared Services.

Any disabled Essbase users or groups do not migrate.

An Essbase user name cannot exist as a group name in Shared Services. If it does, the Essbase user does not migrate.

In Shared Services, if an Essbase application contains multiple databases, the databases must have the same user security access levels. During migration, if a user has different access levels for two databases in the same application, the user is given the more restrictive access level for both databases. In such cases, a warning is sent to the Administrator who ran the migration and the information is logged in the Essbase Server log (`ARBORPATH/essbase.log`). You can also use the MaxL statement **display user** to list instances of multiple database access level changes.

Users and groups are migrated in the following order:

1. Applications are registered with Shared Services.
2. Groups are migrated.
3. Users are migrated.

If a migration fails, the status of the migration depends on the point at which it fails. For example, if the migration fails at step 1, then the total migration fails. If a migration fails at step 2, the result depends on the reason for failure. If a migration fails at step 3, when one or more users fails to migrate, then applications and groups may have been migrated.

Users and groups that fail migration are listed in the Essbase Server log (`ARBORPATH/essbase.log`). You can use the MaxL statements **display user** and **display group** to list users and groups that failed migration and to remigrate all or a selection of these failed users and groups.

When you use Administration Services Externalize Users Wizard to migrate Administration Services users or to remigrate Essbase users that previously failed migration, migration errors are logged in the file that you specify in the wizard, as well as in the Essbase Server log (*ARBORPATH*/`essbase.log`).

If a group fails migration, all users in the group fail migration; you must repair and migrate the group in order for the users to migrate successfully.

The following conditions apply for successful group migration:

- An Essbase group name cannot exist as a user name in Shared Services. If it does, the Essbase group, and all users in the group, do not migrate.
- An Essbase user name cannot exist as a group name in Shared Services. If it does, the Essbase user does not migrate.

If a group exists in both Essbase and Shared Services, the following conditions apply:

- Shared Services cannot contain two groups at different levels in the same hierarchy (an ancestor-child relationship) when the groups exist in Essbase (see Example 2). If it does, the entire migration process fails.
- The Shared Services group cannot contain a user that does not exist in the Essbase group of the same name. If it does, the Essbase group, and all users in the group, do not migrate.
- The Essbase group cannot contain a user that exists in Shared Services, unless the Shared Services user belongs to the Shared Services group of the same name. If it does, the Essbase group, and all users in the group, do not migrate.

The following examples highlight group migration considerations:

Example 1: The groups in this example migrate successfully from Essbase to Shared Services.

Essbase has groups named group 1 and group 2:

```
group 1, group 2
```

Shared Services has two identical groups and also has a group 3, which contains group 1 and group 2:

```
    group 3
       |
group 1, group 2
```

The groups migrate successfully because group 1 and group 2 are at the same level as each other in Shared Services and because Essbase does not have a group 3.

Note:

If group 3 has Administrator (previously known as Supervisor) access to the Essbase Server instance and Essbase group 1 and group 2 have user access, the resulting group 1 and group 2 in Shared Services will have Administrator access.

Example 2: The migration in this example fails because Shared Services contains group 1 and group 2 at different levels.

Essbase has groups named group 1 and group 2:

```
group 1, group 2
```

Shared Services has group 1 and group 2, but group 1 contains group 2:

```
group 1
    |
group 2
```

Example 3: The migration in this example fails because Essbase contains group 1, group 2, and group 3 and Shared Services contains group 3 at a different level from group 1 and group 2.

Essbase has groups named group 1, group 2, and group 3:

```
group 1, group 2, group 3
```

Shared Services has group 1 and group 2, but has a group 3, which contains group 1 and group 2:

```
    group 3
       |
group 1, group 2
```

Continuing to Use Essbase in Native Security Mode

ForEssbase Servers, you can continue to use native authentication if you want to continue managing users and groups as you did in previous releases. In native security mode, you continue to manage users via Administration Services Console. You can continue to create native and external users as you did before.

If you plan to use external authentication in native security mode, you must configure external authenticating through Shared Services. See "Using Shared Services Security for External Authentication in Native Security Mode" on page 633. Shared Services is not required for the custom authentication feature, see the *Oracle Essbase Administration Services Online Help*.

The following options apply:

- Essbase Server and Essbase Administration Server can both run in native security mode. You do not need to install and configure Shared Services if both of the following are true:
  - Essbase and Administration Services are the only Oracle products you are installing.
  - You want Essbase Server and Essbase Administration Server to continue running in native security mode and you do not plan to use external authentication in native security mode.
- Essbase Administration Server can run in Shared Services security mode with Essbase Server running in native security mode. The same rules apply to Essbase Provider Servers.

Note:

If any Essbase Server that you administer from Administration Services Console runs in Shared Services security mode, Essbase Administration Server must also run in Shared Services security mode. You cannot use a combination of Shared Services security mode and native security mode to manage users on an Essbase Server. You must choose one mode for managing all Essbase users on an Essbase Server. Native security mode will not be available in future releases of Essbase.

Understanding Native Security Mode in Essbase

Essbase provides a system for managing access to applications, databases, and other artifacts within Essbase. Using the Essbase security system provides protection in addition to the security available through your local area network.

The Essbase security system addresses a wide variety of database security needs with a multilayered approach to enable you to develop the best plan for your environment. Various levels of permission can be granted to users and groups or defined at the system, application, or database scope. You can apply security in the following ways:

- Users and groups.

  To grant permissions to individual users and groups of users. When higher, these permissions take precedence over minimum permissions defined for applications and databases. Ordinary users have no inherent permissions. Permissions can be granted to users and groups by editing the users and groups or by using the **grant** statement in MaxL DDL (data definition language). See "Granting Permissions to Users and Groups in Native Security Mode" on page 628.

  You can create users who log on using the parameters of an external authentication repository instead of the Essbase password. If you want users to use an outside authentication repository such as LDAP, you must implement the Shared Services security platform and create the Essbase users with a reference to the security platform. See "Using Shared Services Security for External Authentication in Native Security Mode" on page 633.

- Application and database settings.

  To set common permissions for all users of an application or database, you can set minimum permissions that all users can have at each application or database scope. Users and groups with lower permissions than the minimum gain access; users and groups with higher granted permissions are not affected. You can also temporarily disable different kinds of access using application settings. See "Managing Global Security for Applications and Databases in Native Security Mode" on page 634.

- Server-wide settings.

  Create and manage login restrictions for the entire Essbase Server. View and terminate current sessions and requests running on the entire Essbase Server or only on particular applications and databases. See "Managing User Activity on the Essbase Server in Native Security Mode" on page 638.

- Database filters.

  Define database permissions that users and groups can have for particular members, down to the individual data value (cell). See Chapter 39, "Controlling Access to Database Cells."

Table 62 describes security permissions and the tasks that can be performed with those permissions.

Table 62 Description of Essbase Permissions

Permission	Affected Scope	Description
No Access or None	Entire system, application, or database	No inherent access to any users, groups, or data values, unless access is granted globally or by a filter. No Access is the default when creating an ordinary user. Users with No Access permissions can change their passwords.
Read	Database	Ability to read data values.
Write	Database	Ability to read and update data values.
Metaread	Database	Ability to read metadata (dimension and member names) and update data for the corresponding member specification.
Execute (or Calculate)	Entire system, application, database, or single calculation	Ability to calculate, read, and update data values for the assigned scope, using the assigned calculation. Administrators, application managers for the application, and database managers for the database can run calculations without being granted execute access.
Database Manager	Database	Ability to modify outlines, create and assign filters, alter database settings, and remove locks/terminate sessions and requests on the database. A user with Database Manager permission in one database does not necessarily have that permission in another.
Application Manager	Application	Ability to create, delete, and modify databases within the assigned application. Ability to modify the application settings, including minimum permissions, remove locks on the application, terminate sessions and requests on the application, and modify any artifact within the application. You cannot create or delete an application unless you also have been granted the system-level Create/Delete Applications permission. A user with Application Manager permission in one application does not necessarily have that permission in another.
Filter Access	Database	Ability to access specific data and metadata according to the restrictions of a filter assigned to the user or group. The filter definition specifies, for subsets of a database, whether read, write, no access, or metaread is allowed for each subset. A user or group can be granted only one filter per database. Filters can be used in conjunction with other permissions. See Chapter 39, “Controlling Access to Database Cells.”
Create/Delete Applications	Entire system	Ability to create and delete applications and databases within those applications, and control permissions, locks, and resources for applications created. Includes designer permissions for the applications and databases created by this user.

Permission	Affected Scope	Description
Create/Delete Users, Groups	Entire system	Ability to create, delete, edit, or rename all users and groups having equal or lesser permissions than their own.
Administrator	Entire system	Full access to the entire system and all users and groups.

Creating Users and Groups in Native Security Mode

When you create a user or a group in Essbase, you define a security profile. The security profile is where you define the extent of the permissions that users and groups have in dealing with each other and in accessing applications and databases.

If you are using Administration Services, you also must create users on the Essbase Administration Server. See "About Administration Services Users" on page 111.

Creating Users in Native Security Mode

To create a user means to define the user name, password, and permission. You can also specify group membership for the user, and you can specify that the user must change the password at the next login attempt, or that the user name is disabled, preventing the user from logging on.

User names can contain only characters defined within the code page referenced by the ESSLANG variable, and they cannot contain a backslash (\). User names must begin with a letter or a number.

➤ To create a user, use a tool:

Tool	Topic	Location
Administration Services	Creating Users on Essbase Servers	*Oracle Essbase Administration Services Online Help*
MaxL	**create user**	*Oracle Essbase Technical Reference*

For example, to create a user named admin and grant that user Administrator permissions, use the following MaxL statements:

```
create user admin identified by 'password';
grant administrator to admin;
```

Essbase and Planning have the concept of an "application access type" for Essbase and Planning users. For example, when an Essbase user is created using any Essbase administration tool, the user is automatically assigned the application access type "Essbase"; when a Planning user is created using the Planning interface, the user is automatically assigned the application access type "Planning." A user's application access type specifies whether the user has access to Essbase applications only, to Planning applications only, or to both.

➤ To specify application access types for users, use a tool:

Tool	Topic	Location
Administration Services	Setting Application Access Type for Users	*Oracle Essbase Administration Services Online Help*
MaxL	**create user**	*Oracle Essbase Technical Reference*

For information about specifying application access type for Planning users, see Oracle Hyperion Planning, Fusion Edition documentation.

Creating Groups in Native Security Mode

Groups comprise users who share minimum access permissions. Placing users in groups can save you the time of assigning identical permissions to users again and again.

Note:

A member of a group may have permissions beyond those assigned to the group, if permissions are also assigned individually to that user.

The process for creating, editing, or copying groups is the same as that for users, except that there are no group passwords. You define group names and permissions just as you do for users.

Note:

A group name may not contain a backslash (\).

When you create a user, you can assign the user to a group. Similarly, when you create a group, you can assign users to the group. You must define a password for each user; there are no passwords for groups.

➤ To create groups, use a tool:

Tool	Topic	Location
Administration Services	Creating Groups on Essbase Servers	*Oracle Essbase Administration Services Online Help*
MaxL	**create group**	*Oracle Essbase Technical Reference*

Granting Permissions to Users and Groups in Native Security Mode

You can define security permissions for individual users and groups. Groups comprise users who share minimum permissions. Users inherit the permissions of the group and additionally can have access to permissions exceeding those of the group.

Permissions can be granted to users and groups in the following ways:

- Specifying a user or group type upon creation or by editing the user or group. This method specifies system-level permissions that span all applications and databases. For descriptions of the types of users and groups, see "Assigning User and Group Types in Native Security Mode " on page 628.
- Granting permission to access applications or databases. For an explanation of how to grant resource-specific permissions, see "Granting Application and Database Access to Users and Groups in Native Security Mode" on page 629.
- Granting designer permissions to users or groups. For an explanation of situations requiring the granting of designer permissions, see "Granting Designer Permissions to Users and Groups in Native Security Mode" on page 630.

Assigning User and Group Types in Native Security Mode

One way to assign permissions to users and groups is to define user and group types when you create or edit (modify the permissions of) the users and groups.

In Administration Services, users and groups can be created in different ways to specify their system-level permissions. These methods are represented in Administration Services Console as user types. In MaxL, user types do not exist; instead, you grant the permissions after the user is created.

In Administration Services, users can be created with the following types:

- Administrator.

  A user or group with Administrator permission has full access to the entire system and all users and groups. The user who installs Essbase on the server is designated the System Administrator for that server. Essbase requires that at least one user on each server has Administrator permission. Therefore, you cannot delete or downgrade the permission of the last administrator on the server.
- User.

  Users or groups with ordinary permission have no inherent access to any users, groups, or resources. This type of user is the default user.
- Users with Create/Delete Users, Groups permission.

  This type of user or group can create, delete, edit, or rename users and groups with equal or lower permissions only.
- Users with Create/Delete Applications permission.

This type of user or group can create and delete applications and control permissions and resources applicable to those applications or databases they created.

Users with Create/Delete Applications permission cannot create or delete users, but they can manage application-level permission for those applications that they have created. For information about application-level permission, see "Managing Global Security for Applications and Databases in Native Security Mode" on page 634.

For instructions about creating users and groups, see "Creating Users in Native Security Mode" on page 626 and "Creating Groups in Native Security Mode" on page 627.

Granting Application and Database Access to Users and Groups in Native Security Mode

If you need to grant resource-specific permissions to users and groups that are not implied in any user types, you can grant the specific application or database permissions to users when creating or editing them in Administration Services. Using MaxL, you grant the permissions after the user is created by using the **grant** statement.

You can grant or modify user and group application and database permissions from an edit-user standpoint or from an application or database security perspective. The results are the same.

Note:

If a user has insufficient permission to access the data in a database, the value does not show up in queries, or shows up as #NOACCESS.

There is no need to grant permissions to users or groups that are already Administrators—they have full permissions to all resources on the Essbase Server. For a given database, users or groups can also be granted any of the following permissions:

Table 63 Permissions Available at the Database Scope

Database permission	Description
None	Indicates no access to any artifact or data value in a database.
Filter Access	Indicates that data and metadata access is restricted to those filters assigned to the user. (See Chapter 39, "Controlling Access to Database Cells.") The Filter check box grants a filter artifact to a user or group. A user or group can be granted only one filter per database. Selecting this option or any other option except None enables the selection of a filter artifact from the list box.
Read only	Indicates read permission; that is, the ability to retrieve all data values. Report scripts can also be run.
Read-write	Indicates that all data values can be retrieved and updated (but not calculated). The user can run, but cannot modify, Essbase artifacts.
Metaread	Indicates that metadata (dimension and member names) can be retrieved and updated for the corresponding member specification.

Database permission	Description
Calculate	Indicates that all data values can be retrieved, updated, and calculated with the default calculation or any calculation for which the user has been granted permission to execute.
Database Manager	Indicates that all data values can be retrieved, updated, and calculated. In addition, all database-related files can be modified.

➤ To grant or modify application or database permissions for a user or group, use a tool:

Tool	Topic	Location
Administration Services	Managing User/Group Permissions for Applications and Databases	*Oracle Essbase Administration Services Online Help*
MaxL	To grant permissions: **grant** To change the user type or group: **alter user**	*Oracle Essbase Technical Reference*

Granting Designer Permissions to Users and Groups in Native Security Mode

Users and groups can be granted Application Manager or Database Manager permission for particular applications or databases. These permissions are useful for assigning administrative permissions to users who need to be in charge of particular applications or databases but need only ordinary user permissions for other purposes.

You must grant database access to other users if any of the following conditions apply:

- Users have not been granted sufficient user permission to access databases.
- The database in question does not allow users sufficient access through its minimum permission settings.
- Users do not have sufficient access granted to them through filters.

For references to methods you can use to grant Designer permissions to a user or group, see "Granting Application and Database Access to Users and Groups in Native Security Mode" on page 629.

Managing Users and Groups in Native Security Mode

To help manage security between users and groups, the following user-management tasks are available at varying degrees to users with different permissions.

Viewing Users and Groups in Native Security Mode

➤ To view users and groups, use a tool:

Tool	Topic	Location
Administration Services	Viewing Essbase Server Users and Groups	*Oracle Essbase Administration Services Online Help*
MaxL	**display user** or **display group**	*Oracle Essbase Technical Reference*

Editing Users in Native Security Mode

To edit a user means to modify the security profile established when the user was created. For information about changing user passwords, see "Propagating Password Changes in Native Security Mode" on page 640.

➤ To change a password or other user properties, use a tool:

Tool	Topic	Location
Administration Services	Editing Essbase Server User Properties	*Oracle Essbase Administration Services Online Help*
MaxL	**alter user**	*Oracle Essbase Technical Reference*

Editing Groups in Native Security Mode

To edit a group means to modify the security profile established when the group was created.

➤ To view or change group membership, use a tool:

Tool	Topic	Location
Administration Services	Editing Group Properties	*Oracle Essbase Administration Services Online Help*
MaxL	**display user in group** **alter group**	*Oracle Essbase Technical Reference*

Copying an Existing Security Profile in Native Security Mode

An easy way to create a user with the same permissions as another user is to copy the security profile of an existing user. The new user is assigned the same user type, group membership, and application/database access as the original user.

You can also create new groups by copying the security profile of an existing group. The new group is assigned the same group type, user membership, and application access as the original group.

You can copy users and groups on the same Essbase Server or from one Essbase Server to another, according to your permissions. You can also migrate users and groups across servers along with an application. See "Copying Users" in the *Oracle Essbase Administration Services Online Help*.

To copy a user or group, you duplicate the security profile of an existing user or group and give it a new name, which saves you the time of reassigning permissions when you want them to be identical.

Note:

Copying removes any security permissions that the creator does not have from the copy. For example, a user with Create/Delete Users permission cannot create an administrator by copying the profile of an existing administrator.

➤ To create a user or group by copying the security profile of an existing user or group, use a tool:

Tool	Topic	Location
Administration Services	Copying Essbase Server Users Copying Groups	*Oracle Essbase Administration Services Online Help*
MaxL	**create user** **create group**	*Oracle Essbase Technical Reference*

Deleting Users and Groups in Native Security Mode

➤ To delete users and groups, use a tool:

Tool	Topic	Location
Administration Services	Deleting Essbase Server Users Deleting Groups	*Oracle Essbase Administration Services Online Help*
MaxL	**drop user** **drop group**	*Oracle Essbase Technical Reference*

Renaming Users and Groups in Native Security Mode

➤ To rename users and groups, use a tool:

Note:

A group name may not contain a backslash (\).

Tool	Topic	Location
Administration Services	Renaming Essbase Server Users Renaming Groups	*Oracle Essbase Administration Services Online Help*
MaxL	**alter user** **alter group**	*Oracle Essbase Technical Reference*

Using Shared Services Security for External Authentication in Native Security Mode

External authentication means that the user login information needed by Essbase is maintained in a central authentication directory, such as Oracle Internet Directory or Lightweight Directory Access Protocol (LDAP) Directory.

An authentication directory is a centralized store of user information such as login names and passwords, and other corporate information. The repository functions like a telephone directory. The authentication directory probably contains much more than user names and passwords; for example, it may include e-mail addresses, employee IDs, job titles, access rights, and telephone numbers. It may also contain artifacts other than users; for example, it may contain information about corporate locations or other entities.

To use Shared Services security for external authentication in native security mode, you must install and configure Shared Services:

- Register Essbase with Shared Services.

  See the *Oracle Hyperion Enterprise Performance Management System Installation and Configuration Guide*
- Configure user directories for Essbase.

  See *Oracle Hyperion Enterprise Performance Management System Security Administration Guide*

➤ To manage external authentication of users using Administration Services, see “Managing External Authentication” in the *Oracle Essbase Administration Services Online Help*.

Managing Global Security for Applications and Databases in Native Security Mode

In addition to granting permissions to users and groups, you can change security settings for entire applications and databases and their related files and resources. Application and database security settings enable you to manage connections and create a lowest-common-security profile for the applications and databases.

Defining Application Settings in Native Security Mode

You can define permissions and other security settings that apply to applications by changing the application settings. The settings you define for the application affect all users, unless they have higher permissions granted to them at the user level.

Only users with Administrator permission (or Application Manager permission for the application) can change application settings.

To define settings for an application, see the next two sections.

Setting General Application Connection Options in Native Security Mode

The following application settings are available:

- A brief description of the application
- A timeout setting for locks
- Options that control application loading, command processing, connections, updates, and security
- Settings that define the minimum permissions to the application. See “Setting Application and Database Minimum Permissions in Native Security Mode” on page 637.

The following settings are available for various levels of application security. For information about how and when disabling these settings takes effect, see Table 64.

- Allow Application to Start

  When disabled, prevents all users from starting the application directly or as a result of operations that would start the application; for example, attempting to change application settings or create databases. By default, the application is not prevented from starting.

- Start When Essbase Server Starts

  When enabled, the application starts automatically whenever the Essbase Server starts. By default, the application does not start when the Essbase Server starts.

- Allow commands

  When unchecked, prevents users from making requests to databases in the application, including non-data-specific requests such as viewing database information or changing database settings. Administrators are affected by this setting as a safety mechanism to prevent accidental changes to databases during maintenance operations. By default, commands are enabled.

- Allow connects

  When unchecked, prevents users with a permission lower than Application Manager for that application from making connections to databases within the application which require the databases to be started. By default, connections to databases are allowed.

- Allow updates

  When unchecked, prevents modification to on-disk database structures; for example, any operation that might have an effect on the data. This restriction does not include outline operations. To block metadata updates, set the database to read-only mode or uncheck Allow Commands and/or Allow Connects. By default, updates are enabled.

- Enable Security

  When unchecked, Essbase ignores all security settings in the application and treats all users as Application Managers. By default, security is enabled.

Table 64 describes when the implementation of protective application settings takes effect, how long the effects last, and which users are affected.

Table 64 Scope and Persistence of Application-Protection Settings

Disabled Application Setting	When the Disabled Setting Takes Effect	Which Users are Affected by the Disabled Setting	Persistence of the Disabled Setting
Allow Users to Start Application	Immediately	All users, including administrators. Users currently logged on and users who log on later.	The application cannot be started until an administrator re-enables the startup setting.
Start Application When Essbase Server Starts	Immediately	All users.	The application will not start with Essbase Server unless an administrator enables it.
Allow Commands	Immediately	All users, including administrators. Users currently logged on and users who log on later.	Commands are disabled until any of the following actions occur: 1. The administrator who disabled commands logs off. 2. The application is stopped and restarted. 3. An administrator re-enables commands.
Allow Connects	Immediately, except that disabling connections does not affect users who already have databases loaded.	Users with permissions lower than Application Manager.	Connections are disabled until any of the following actions occur:

Disabled Application Setting	When the Disabled Setting Takes Effect	Which Users are Affected by the Disabled Setting	Persistence of the Disabled Setting
		Users currently logged on and users who log on later. Users already connected to the database are not affected.	1. The application is stopped and restarted. 2. An administrator re-enables connections.
Allow Updates	Immediately	All users, including administrators. Users currently logged on and users who log on later.	Updates are disabled until any of the following actions occur: 1. The administrator who disabled updates logs off. 2. The application is stopped and restarted. 3. An administrator re-enables updates.
Enable Security	Immediately	All users, including administrators. Users currently logged on and users who log on later.	Security is disabled until a user re-enables security.

➤ To change application settings, use a tool:

Tool	Topic	Location
Administration Services	Setting Application Properties	*Oracle Essbase Administration Services Online Help*
MaxL	**alter application**	*Oracle Essbase Technical Reference*

Note:

If performing maintenance operations that require disabling *commands* or *updates*, make those maintenance operations within the same session as the one in which the setting was disabled.

If you disable commands or updates in a MaxL script, be aware that the end of the script constitutes the end of the session. Calling a nested MaxL or ESSCMD script from the current MaxL script also constitutes the end of the session.

If you disable commands or updates in an ESSCMD script, the end of the script constitutes the end of the session, but calling a nested ESSCMD script from the current ESSCMD script does *not* constitute the end of the session.

Caution!

Never power down or reboot your client computer when you have cleared any Allow settings. Always log off from the server correctly. Improper shutdown can cause the application to become inaccessible, which requires a full application shutdown and restart.

If a power failure or system problem causes Essbase Server to improperly disconnect from the Essbase client, and the application is no longer accessible, you must shut down and restart the application. See "Starting and Stopping Applications" on page 701.

Setting Application and Database Minimum Permissions in Native Security Mode

Minimum database access permissions can be specified at the application or database level. If specified for an application, minimum database access permissions apply to all databases within the application. When a minimum permission is set to a level higher than None (or No Access) for an application or database, all users inherit that permission to access the database or databases.

For example, if an application has read permission assigned as the minimum database access level, all users can read any database within that application, even if their individual permissions do not include read access. Similarly, if a database has a minimum permission setting of None, only users with sufficient granted permissions (granted directly or implied by filters or group membership) can gain access to the database.

Users with Administrator, Application Manager, or Database Manager permissions are not affected by minimum permission settings applied to applications or databases they own. Administrators have full access to all resources, and Application Managers and Database Managers have full access for their applications or databases.

Users and groups with lower than the minimum permissions inherit at least the minimum permissions for any applications or databases.

Changes to the minimum permission settings for applications affect only those databases that have lower minimums. In other words, settings defined at a lower level take precedence over more global settings.

The permissions listed in Table 65 are available as minimum settings for applications and databases. Databases of an application inherit the permissions of the applications whenever the application permissions are set higher than those of the database.

Table 65 Minimum Permission Settings Available for Applications and Databases

Permission	Description
None	Specifies that no minimum permission has been defined for the application or database. None is the default global permission for newly created applications and databases.
Read	Specifies read-only access to any artifact or data value in the application or database. Users can view files, retrieve data values, and run report scripts. Read access does not permit data-value updates, calculations, or outline modifications.
Write	Specifies Update access to any data value in the databases of the application, or in one database. Users can view Essbase files, retrieve and update data values, and run report scripts. Write access does not permit calculations or outline modifications.

Permission	Description
Metaread	Gives read access to the specified members but hides data for their ancestors and hides data and metadata for their siblings.
Calculate	Specifies Calculate and update access to any data value in the databases of the application, or in one database. Users can view files, retrieve, update, and perform calculations based on data values, and run report and calculations scripts. Calculate access does not permit outline modifications.
Designer (for Application or Database)	Specifies Calculate and update access to any data value in the databases of the application, or in one database. In addition, Designer permission enables users to view and modify the outline and files, retrieve, update, and perform calculations based on data values, and run report and calculation scripts.

Note:

Although any user with a minimum of read access to a database can start the database, only an Administrator, a user with Application Manager permission for the application, or a user with Database Manager permission for the database can stop the database.

➤ To set minimum permissions for an application, use a tool:

Tool	Topic	Location
Administration Services	Setting Minimum Permissions for Applications	*Oracle Essbase Administration Services Online Help*
MaxL	**alter application**	*Oracle Essbase Technical Reference*

➤ To set minimum permissions for a database, use a tool:

Tool	Topic	Location
Administration Services	Setting Minimum Permissions for Databases	*Oracle Essbase Administration Services Online Help*
MaxL	**alter database**	*Oracle Essbase Technical Reference*

Managing User Activity on the Essbase Server in Native Security Mode

This topic explains how to manage the activities of users connected to the Essbase Server. The security concepts explained in this section are session and request management, lock management, connection management, and password and user name management. For information about managing security for partitioned databases, see Chapter 14, "Designing Partitioned Applications."

Disconnecting Users and Terminating Requests in Native Security Mode

The security system lets you disconnect a user from the Essbase Server in order to perform maintenance tasks.

To view sessions, disconnect sessions, or terminate requests, you must have Administrator permission or Application Manager permission for the specified application. You can view or terminate only sessions or requests for users with permissions equal to or lesser than your own.

A *session* is the time between login and logout for a user connected to Essbase Server at the system, application, or database scope. A user can have multiple sessions open at any time; for example, a user may be logged on to different databases. If you have the appropriate permissions, you can log off sessions based on any criteria you choose; for example, an administrator can log off a user from all databases or from one database.

A *request* is a query sent to Essbase Server by a user or by another process; for example, a default calculation of a database, or a restructuring of the database outline. Each session can process only one request at a time.

Note:

You cannot terminate a restructure process. If you attempt to terminate it, a "command not accepted" error is returned, and the restructure process is not terminated.

➤ To disconnect a session or request using Administration Services, see “Disconnecting User Sessions and Requests” in the *Oracle Essbase Administration Services Online Help*.

➤ To disconnect a session or request using MaxL, use `alter system kill request` or `alter system logout session`. See the *Oracle Essbase Technical Reference*.

Managing User Locks in Native Security Mode

Spreadsheet Add-in users can interactively send data from a spreadsheet to the server. To maintain data integrity while providing multiple-user concurrent access, Essbase enables users to lock data for the purpose of updating it. Users who want to update data must first lock the records to prevent other users from trying to change the same data.

Occasionally, you may need to force an unlock operation. For example, if you attempt to calculate a database that has active locks, the calculation must wait when it encounters a lock. By clearing the locks, you allow the calculation to resume.

Only Administrators can view users holding locks and remove their locks.

➤ To view or remove locks, use a tool:

Tool	Topic	Location
Administration Services	Viewing Data Locks and Unlocking Data	*Oracle Essbase Administration Services Online Help*
MaxL	**drop lock**	*Oracle Essbase Technical Reference*

Managing Passwords and User Names in Native Security Mode

You can place limitations on the number of login attempts users are allowed, on the number of days users may not use Essbase before becoming disabled from the server, and on the number of days users are allowed to have the same passwords. Only system administrators (users with Administrator permission) can access these settings. The limitations apply to all users on the server and are effective upon clicking OK.

Note:

If you later change the number of unsuccessful login attempts allowed, Essbase resets the count for all users. For example, if the setting was 15 and you changed it to 20, users are allowed 20 *new* attempts. If you changed the setting to 2, a user who had exceeded that number when the setting was 15 is *not* locked out. The count returns to 0 for each change in settings.

➤ To place limitations on users, use a tool:

Tool	Topic	Location
Administration Services	Managing Password Longevity Disconnecting Users Automatically	*Oracle Essbase Administration Services Online Help*
MaxL	**alter system** **alter user**	*Oracle Essbase Technical Reference*

Propagating Password Changes in Native Security Mode

You can change a user's password and propagate the new password to other Essbase Servers. You need Create/Delete Users and Groups permissions for both the source and the target servers. The user whose password you are changing must exist on the target servers, and the target servers must be running.

If you use Administration Services to change a user's Essbase Server password, and if the user is also an Administration Services user, the user's Administration Services user properties are updated automatically. The user's Administration Services password is not affected. See "Changing Passwords for Essbase Server Users" in the *Oracle Essbase Administration Services Online Help*.

➤ To change a user's Essbase Server password and propagate the new password to other Essbase Servers, see "Propagating Passwords Across Servers" in the *Oracle Essbase Administration Services Online Help*.

Viewing and Activating Disabled User Names in Native Security Mode

You can prevent a user from logging in to an Essbase Server by disabling the user name at the Essbase Server level. A user name is disabled automatically when the user exceeds limitations specified, or a user name can be disabled manually for individual users. For more information about limitations that cause user names to become disabled automatically, see "Managing Passwords and User Names in Native Security Mode" on page 640.

Administration Services provides a Disabled Usernames window that enables you to view and activate all user names that have been disabled for an Essbase Server. Only users with at least Create/Delete User permission can view or reactivate disabled user names.

➤ To disable a user name manually, use a tool:

Tool	Topic	Location
Administration Services	Disabling Usernames	*Oracle Essbase Administration Services Online Help*
MaxL	**alter user**	*Oracle Essbase Technical Reference*

➤ To view or activate currently disabled user names, use a tool:

Tool	Topic	Location
Administration Services	Viewing or Activating Disabled Usernames	*Oracle Essbase Administration Services Online Help*
MaxL	**alter user**	*Oracle Essbase Technical Reference*

Understanding the essbase.sec Security File

The contents of the `essbase.sec` security file are encrypted. When you back up `essbase.sec`, the contents of the `essbase.bak` backup file are also encrypted. The backup procedure for Essbase security information depends on whether you are using Essbase in native security mode or in Shared Services security mode.

To review the contents of the `essbase.sec` file, you can export the contents to a readable, text-file format. See "Exporting the Security File" on page 646.

Security File Backups in Native Security Mode

When Essbase is in native security mode, all information about users, groups, passwords, permissions, filters, applications, databases, and their corresponding directories is stored in the Essbase security file (`essbase.sec`) in the *`ESSBASEPATH`*`/bin` directory. See "Backing Up the Security File" on page 643.

Security File Backups in Shared Services Security Mode

When Essbase is in Shared Services security mode, some security information is stored by Shared Services and/or by the external user directories, and some security information is stored in the Essbase security file (`essbase.sec`).

When Essbase is in Shared Services security mode, in addition to backing up the Essbase security file (`essbase.sec`), you must follow backup procedures for Shared Services and for any external authentication directories.

The following information is stored by Shared Services or by the external user directories:

- Users
- Groups
- Passwords
- User and group role information for applications

For information on backup procedures, see the *Oracle Hyperion Enterprise Performance Management System Backup and Recovery Guide* and the documentation for the appropriate external user directories.

The following information is stored in the Essbase security file (`essbase.sec`) in the *`ESSBASEPATH`*`/bin` directory:

- Calculation script access
- Filter access
- Application access type
- Application and database properties, including substitution variables and DISKVOLUMES settings (block storage databases only).

See "Backing Up the Security File" on page 643.

Caution!

Back up the Essbase security file and Shared Services simultaneously.

Note:

Essbase automatically creates a backup of the security file before and after migration (`essbase.bak_preUPM` and `essbase.bak_postUPM`). See "Migrating Essbase to Shared Services" on page 619.

Security Information Recovery in Shared Services Security Mode

If a discrepancy occurs between the security information in Shared Services and the security information in the Essbase security file, the type of discrepancy determines whether Shared Services information or the Essbase security file information takes precedence.

User and Group Information

Shared Services takes precedence for user and group information. User and group information can be restored using Shared Services and external user directory backup and recovery procedures (see the *Oracle Hyperion Enterprise Performance Management System Backup and Recovery Guide*).

Note:

When recovering user and group information, any associations to filters, calculation scripts, and application access type are lost if the Shared Services backup does not have the information. If the Essbase security backup file does not have the information, the filters and calc scripts themselves are lost (not just the associations to the users or groups).

Application Information

Essbase takes precedence for application and database information. If an application is deleted from Shared Services, the application is still available in Essbase. You must reregister the application with Shared Services. If an application is deleted in Essbase, it is automatically deleted from Shared Services.

➤ To reregister an application with Shared Services, use a tool:

Tool	Topic	Location
Administration Services	Reregistering an Application With Shared Services	*Oracle Essbase Administration Services Online Help*
MaxL	**alter application**	*Oracle Essbase Technical Reference*

Backing Up the Security File

Each time you successfully start Essbase Server, two backup copies of the security file are created —`essbase.bak` and `essbase.bak_startup`.

The `essbase.bak_startup` file is updated only at Essbase Server startup. You cannot update it any other time.

You can update `essbase.bak` more often using one of the following methods:

- Manually compare `essbase.bak` to the security file at any time and update it if necessary.

- Specify an interval at which Essbase automatically compares `essbase.bak` to the security file and updates`essbase.bak`, if necessary. See "Changing Security Backup File Comparison Frequency" on page 644.

➤ To update the `essbase.bak` file, use a tool:

Tool	Topic	Location
Administration Services	Updating the Security Backup File	*Oracle Essbase Administration Services Online Help*
MaxL	**alter system sync security backup**	*Oracle Essbase Technical Reference*

If you attempt to start Essbase Server and cannot get a password prompt or your password is rejected, no backup files are created. You can restore from the last successful startup by copying `essbase.bak` to `essbase.sec`. Both files are in the *`ESSBASEPATH`*`/bin` directory where you installed Essbase. If you are using Essbase in Shared Services security mode, you must also restore the latest backups from Shared Services and any external user directories.

Caution!

If Essbase stops running unexpectedly for any reason, such as a freeze, abnormal shutdown, or as the result of terminating a process, do not restart Essbase Server until you copy the backup file (`essbase.bak`) to the security file (`essbase.sec`). If you do not perform the copy first, Essbase may replace the `essbase.bak` file with the corrupted `essbase.sec` file.

In the event that the `essbase.bak` file is destroyed or lost, you can restore the security file using the `essbase.bak_startup` file by copying `essbase.bak_startup` to the security file (`essbase.sec`).

Changing Security Backup File Comparison Frequency

Essbase updates the `essbase.bak` security backup file if it does not match the `essbase.sec` security file. By default, Essbase compares the security backup file to the security file at specified intervals instead of only when Essbase Server starts.

See "Updating the Security Backup File" in the *Oracle Essbase Administration Services Online Help* for information about updating the security backup file anytime.

Consider the following facts before changing the interval value:

- In Administration Services, the same check box manages how often the security backup file is checked against the security file and how often user inactivity is checked.
- The default value is five minutes, the recommended setting to ensure that the security backup file is checked frequently enough to capture security changes. Five minutes is also the recommended value for the inactivity check.

- If you set the value to zero, the inactivity check is disabled, and the `essbase.bak` file is compared to `essbase.sec` every five minutes (and updated if necessary).
- Enter a larger value if your security file does not need to be updated frequently. Enter a smaller value if performance is not an issue.

➤ To change the frequency of backup file comparisons, use a tool:

Tool	Topic	Location
Administration Services	Enter the time interval in the Check for inactivity every option of the Security tab when you edit Essbase Server properties.	*Oracle Essbase Administration Services Online Help*
MaxL	**alter system sync security_backup**	*Oracle Essbase Technical Reference*

Caution!

If Essbase stops running unexpectedly for any reason, such as a freeze, abnormal shutdown, or as the result of terminating a process, do not restart Essbase Server until you copy the backup file `essbase.bak` to the security file `essbase.sec`. If you do not perform the copy first, when Essbase Server starts, Essbase notes that `essbase.sec` is corrupt, creates an empty security file, and copies it to `essbase.bak`, destroying the backup of your security information.

Managing Security-File Fragmentation

Changing or deleting the Essbase security entities can cause fragmentation in the security file (`essbase.sec`): filters, users, groups, applications, databases, substitution variables, disk volumes, passwords, and other Essbase artifacts. Too much fragmentation in files can slow down security-related performance.

Essbase compacts the security file automatically each time the Agent is stopped. You can check the fragmentation status of the security file and, if desired, you can compact the security file without stopping the Agent.

Displaying the Security File Fragmentation Status

The fragmentation status of the security file is displayed as a percent.

➤ To display the fragmentation status of the security file, see the **display system** MaxL statement with the **security file fragmentation_percent** grammar in the *Oracle Essbase Technical Reference*.

Compacting the Security File While the Agent is Running

Besides manually compacting the security file, you can use the SECURITYFILECOMPACTIONPERCENT configuration setting to define a percentage of fragmentation that triggers compaction automatically.

➤ To compact the security file without stopping the Agent, use a tool:

Tool	Topic	Location
Agent	COMPACT	Enter the Agent command at the command prompt in the Essbase Server console window.
MaxL	**alter system compact security file**	*Oracle Essbase Technical Reference*
`essbase.cfg`	SECURITYFILECOMPACTIONPERCENT	*Oracle Essbase Technical Reference*

Note:

Compacting the security file while the Agent is running slows down Agent activity until the operation is completed, which could take a few minutes.

Exporting the Security File

An Essbase Administrator can export the contents of the `essbase.sec` file for an Essbase Server instance to a readable text file format, which is useful for review purposes.

Caution!

When exporting `essbase.sec`, follow your company's security procedures to ensure the integrity of the data.

The export security file command, which can be run from Administration Services Console or as a MaxL statement, is run against the Essbase Server session for which you are currently logged in. The Essbase Server session can be run as a service.

➤ To export `essbase.sec`, use a tool:

Tool	Topic	Location
Administration Services	Exporting the Security File	*Oracle Essbase Administration Services Online Help*
MaxL	**export security_file**	*Oracle Essbase Technical Reference*

Note:

The DUMP agent command is similar to the export security file command, except that the DUMP command cannot be run against an Essbase Server run as a service. See Table 70 on page 689.

39 Controlling Access to Database Cells

In This Chapter

Introduction

When security levels defined for applications, databases, users, and groups are insufficient, Essbase security filters give you more specific control. Filters enable you to control access to individual data within a database by defining what kind of access is allowed to which parts of the database, and to whom these settings apply.

If you have Administrator permissions, you can define and assign any filters to any users or groups. Filters do not affect you.

If you have Create/Delete Applications permissions, you can assign and define filters for applications that you created.

If you have Application Manager or Database Manager permissions, you can define and assign filters within your applications or databases.

Understanding How Filters Define Permissions

Filters control security access to data values, or cells. You create filters to accommodate security needs for specific parts of a database. When you define a filter, you designate restrictions on particular database cells. When you save the filter, you give it a unique name to distinguish it from other filters, and the server stores it in `essbase.sec`, the security file. You can then assign the filters to any users or groups on the server.

For example, a manager designs a filter named RED and associates it with a database to limit access to cells containing profit information. The filter is assigned to a visiting group called REVIEWERS, so that they can read, but cannot alter, most of the database; they have no access to Profit data values.

Filters comprise one or more access settings for database members. You can specify the following access levels and apply them to data ranging from a list of members to one cell.

Access Level	Description
None	No data can be retrieved or updated for the specified member list.
Read	Data can be retrieved but not updated for the specified member list.
Write	Data can be retrieved and updated for the specified member list.
Metaread	Metadata (dimension and member names) can be retrieved and updated for the corresponding member specification.

Note:

The metaread access level overrides all other access levels. If additional filters for data are defined, they are enforced within any defined metaread filters.If you have assigned a metaread filter on a substitution variable and then try to retrieve the substitution variable, an unknown member error occurs, but the value of the substitution variable gets displayed. This is expected behavior.Metadata security cannot be completely turned off in partitions. Therefore, do not set metadata security at the source database; otherwise, incorrect data may result at the target partition.When drilling up or retrieving on a member that has metadata security turned on and has shared members in the children, an unknown member error occurs because the original members of the shared members have been filtered. To avoid this error, give the original members of the shared members metadata security access.

Any cells that are not specified in the filter definition inherit the database access level. Filters can, however, add or remove access assigned at the database level, because the filter definition, being more data-specific, indicates a greater level of detail than the more general database access level.

Data values not covered by filter definitions default first to the access levels defined for users and, when Essbase is in native security mode, second to the global database access levels.

Calculation access is controlled by permissions granted to users and groups. Users who have calculate access to the database are not blocked by filters—they can affect all data elements that the execution of their calculations would update. When Essbase is in native security mode, calculation access is also controlled by minimum global permissions for the application or database.

Creating Filters

You can create a filter for each set of access restrictions you need to place on database values. You need not create separate filters for users with the same access needs. After you have created a filter, you can assign it to multiple users or groups of users. However, only one filter per database can be assigned to a user or group.

Note:

If you use a calculation function that returns a set of members, such as children or descendants, and it evaluates to an empty set, the security filter is not created. An error is written to the application log stating that the region definition evaluated to an empty set.

Before creating a filter, perform the following actions:

- Connect to the server and select the database associated with the filter.
- Check the naming rules for filters in Appendix A, "Limits."

➤ To create a filter, use a tool:

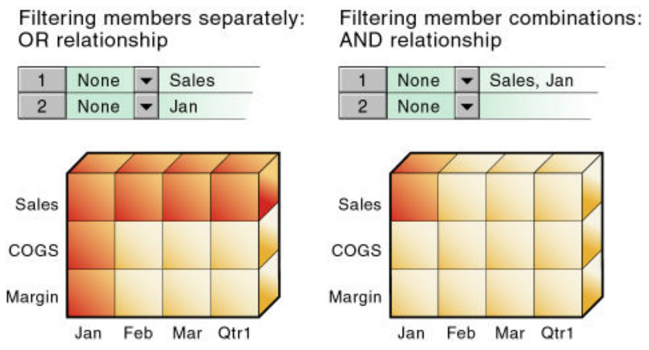
Tool	Topic	Location
Administration Services	Creating or Editing Filters	*Oracle Essbase Administration Services Online Help*
MaxL	**create filter**	*Oracle Essbase Technical Reference*

Filtering Members Versus Filtering Member Combinations

Figure 147 on page 649 illustrates different ways to control access to database cells. Data can be protected by filtering entire members or by filtering member combinations.

- Filtering members separately affects whole regions of data for those members.
- Filtering member combinations affects data at the member intersections.

Figure 147 How Filters Affect Data AND/OR Relationships



Note:

Filtering on member combinations (AND relationship) does not apply to metaread. Metaread filters each member separately (OR relationship).

Filtering Members Separately

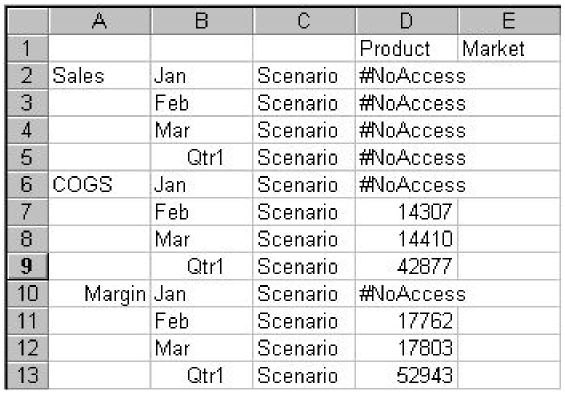
To filter all the data for one or more members, define access for each member on its own row in Filter Editor. Filter definitions on separate rows of a filter are treated with an OR relationship.

For example, to block access to Sales or Jan, assume that user KSmith is assigned this filter:

Access	Member Specification
None	Sales
None	Jan

The next time user KSmith connects to Sample.Basic, she has no access to data values for the member Sales or for the member Jan. Her spreadsheet view of the profit margin for Qtr1:

Figure 148 Results of Filter Blocking Access to Sales or Jan



	A	B	C	D	E
1				Product	Market
2	Sales	Jan	Scenario	#NoAccess	
3		Feb	Scenario	#NoAccess	
4		Mar	Scenario	#NoAccess	
5		Qtr1	Scenario	#NoAccess	
6	COGS	Jan	Scenario	#NoAccess	
7		Feb	Scenario	14307	
8		Mar	Scenario	14410	
9		Qtr1	Scenario	42877	
10	Margin	Jan	Scenario	#NoAccess	
11		Feb	Scenario	17762	
12		Mar	Scenario	17803	
13		Qtr1	Scenario	52943	

All data for Sales is blocked from view, as well as all data for January, inside and outside of the Sales member. Data for COGS (Cost of Goods Sold), a sibling of Sales and a child of Margin, is available, with the exception of COGS for January.

Filtering Member Combinations

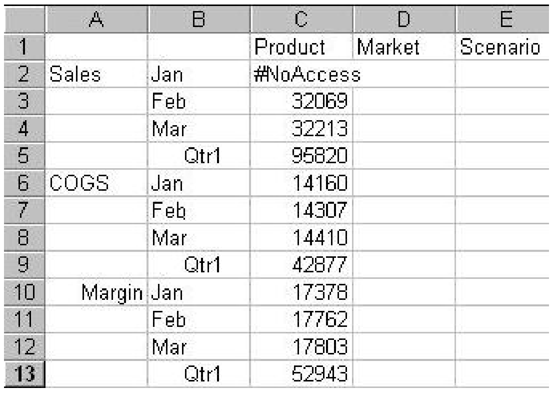
To filter data for member combinations, define the access for each member combination using a row in Filter Editor. In filter definitions, two member sets separated by a comma are treated as union of those two member sets (an AND relationship).

For example, assume that user RChinn is assigned this filter:

Access	Member Specification
None	Sales, Jan

The next time user RChinn connects to Sample.Basic, she has no access to the data value at the intersection of members Sales and Jan. Her spreadsheet view of the profit margin for Qtr1:

Figure 149 Results of Filter Blocking Access to Sales, Jan



	A	B	C	D	E
1			Product	Market	Scenario
2	Sales	Jan	#NoAccess		
3		Feb	32069		
4		Mar	32213		
5		Qtr1	95820		
6	COGS	Jan	14160		
7		Feb	14307		
8		Mar	14410		
9		Qtr1	42877		
10	Margin	Jan	17378		
11		Feb	17762		
12		Mar	17803		
13		Qtr1	52943		

Sales data for January is blocked from view. However, Sales data for other months is available, and non-Sales data for January is available.

Filtering Using Substitution Variables

Substitution variables enable you to more easily manage information that changes regularly. Each substitution variable has an assigned name and value. The Database Manager can change the value anytime. Where a substitution variable is specified in a filter, the substitution variable value at that time is used.

For example, if you want a group of users to see data only for the current month, you can set up a substitution variable named CurMonth and define a filter (MonthlyAccess) wherein you specify access, using `&CurMonth` for the member name. Using an ampersand (&) at the beginning of a specification identifies it as a substitution variable instead of a member name to Essbase. Assign the MonthlyAccess filter to the appropriate users.

Each month, you need to change only the value of the CurMonth substitution variable to the member name for the current month, such as Jan, Feb, and so on. The new value will apply to all assigned users.

See "Using Substitution Variables" on page 120.

Filtering with Attribute Functions

You can use filters to restrict access to data for base members sharing a particular attribute. To filter data for members with particular attributes defined in an attribute dimension, use the attribute member in combination with the @ATTRIBUTE function or the @WITHATTR function.

Note:

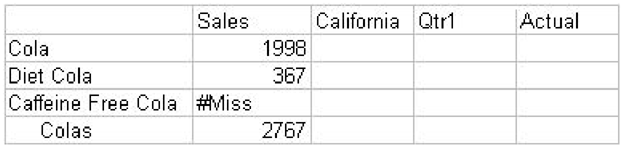
@ATTRIBUTE and @WITHATTR are member set functions. Most member set functions can be used in filter definitions.

For example, assume that user PJones is assigned this filter:

Access	Member Specification
None	@ATTRIBUTE(“Caffeinated_False”)

The next time user PJones connects to Sample.Basic, he has no access to the data values for any base dimension members associated with Caffeinated_False. His spreadsheet view of first-quarter cola sales in California:

Figure 150 Results of Filter Blocking Access to Caffeine-free Products



	Sales	California	Qtr1	Actual
Cola	1998			
Diet Cola	367			
Caffeine Free Cola	#Miss			
Colas	2767			

Sales data for Caffeine Free Cola is blocked from view. Note that Caffeine Free Cola is a base member, and Caffeinated_False is an associated member of the attribute dimension Caffeinated (not shown in the above spreadsheet view).

Metadata Filtering

Metadata filtering provides an additional layer of security in addition to data filtering. With metadata filtering, an administrator can remove outline members from a user's view, providing access only to those members that are of interest to the user.

When a filter is used to apply MetaRead permission on a member,

1. Data for all ancestors of that member are hidden from the filter user’s view.
2. Data *and* metadata (member names) for all siblings of that member are hidden from the filter user’s view.

Managing Filters

You can perform the following actions on filters: viewing, editing, copying, renaming, and deleting.

Viewing Filters

➤ To view a list of filters, use a tool:

Tool	Topic	Location
Administration Services	Creating or Editing Filters	*Oracle Essbase Administration Services Online Help*
MaxL	**display filter**	*Oracle Essbase Technical Reference*

Tool	Topic	Location
ESSCMD	LISTFILTERS	*Oracle Essbase Technical Reference*

Editing Filters

➤ To edit a filter, use a tool:

Tool	Topic	Location
Administration Services	Creating or Editing Filters	*Oracle Essbase Administration Services Online Help*
MaxL	**create filter**	*Oracle Essbase Technical Reference*

Copying Filters

You can copy filters to applications and databases on any Essbase Server, according to your permissions. You can also copy filters across servers as part of application migration.

➤ To copy a filter, use a tool:

Tool	Topic	Location
Administration Services	Copying Filters	*Oracle Essbase Administration Services Online Help*
MaxL	**create filter**	*Oracle Essbase Technical Reference*
ESSCMD	COPYFILTER	*Oracle Essbase Technical Reference*

Renaming Filters

➤ To rename a filter, use a tool:

Tool	Topic	Location
Administration Services	Renaming Filters	*Oracle Essbase Administration Services Online Help*
MaxL	**create filter**	*Oracle Essbase Technical Reference*
ESSCMD	RENAMEFILTER	*Oracle Essbase Technical Reference*

Deleting Filters

➤ To delete a filter, use a tool:

Tool	Topic	Location
Administration Services	Deleting Filters	*Oracle Essbase Administration Services Online Help*
MaxL	**drop filter**	*Oracle Essbase Technical Reference*

Assigning Filters

After you define filters, you can assign them to users or groups, which lets you manage multiple users who require the same filter settings. Modifications to the definition of a filter are automatically inherited by users of that filter.

Filters do not affect users who have the Administrator role. Only one filter per database can be assigned to a user or group.

Assigning Filters in Shared Services Security Mode

In Oracle's Hyperion® Shared Services security mode, you assign filters through Oracle's Hyperion® Shared Services Console.

➤ To assign a filter to a user or group, see "Assigning Database Calculation and Filter Access" on page 616.

Assigning Filters in Native Security Mode

➤ To assign a filter to a user or group, see "Assigning Filters" in the *Oracle Essbase Administration Services Online Help*.

Overlapping Filter Definitions

If a filter contains rows that have overlapping member specifications, the inherited access is set by the following rules, listed in order of precedence:

1. A filter that defines a more detailed dimension combination list takes precedence over a filter with less detail.
2. If the preceding rule does not resolve the overlap conflict, the highest access level among overlapping filter rows is applied.

For example, this filter contains overlap conflicts:

Access	Member Specification
Write	Actual
None	Actual
Read	Actual, @IDESCENDANTS(“New York”)

The third specification defines security at a greater level of detail than the other two. Therefore, read access is granted to all Actual data for members in the New York branch.

Because write access is a higher access level than none, the remaining data values in Actual are granted write access.

All other cells, such as Budget, are accessible according to the minimum database permissions.

If you have write access, you also have read access.

Note:

Changes to members in the database outline are not reflected automatically in filters. You must manually update member references that change.

Overlapping Metadata Filter Definitions

You should define a MetaRead filter using multiple rows only when the affected member set in any given row (the metaread members and their ancestors) has no overlap with MetaRead members in other rows. It is recommended that you specify one dimension per row in filters that contain MetaRead on multiple rows. However, as long as there is no overlap between the ancestors and MetaRead members, it is still valid to specify different member sets of one dimension into multiple MetaRead rows.

For example, in Sample Basic, the following filter definition has overlap conflicts:

Access	Member Specification
MetaRead	California
MetaRead	West

In the first row, applying MetaRead to California has the effect of allowing access to California but blocking access to its ancestors. Therefore, the MetaRead access to West is ignored; users who are assigned this filter will have no access to West.

If you wish to assign MetaRead access to West as well as California, then the appropriate method is to combine them into one row:

Access	Member Specification
MetaRead	California,West

Overlapping Access Definitions

When the access rights of user and group definitions overlap, the following rules, listed in order of precedence, apply:

1. An access level that defines a more detailed dimension combination list takes precedence over a level with less detail.
2. If the preceding rule does not resolve the overlap conflict, the highest access level is applied.

Example 1:

User Fred is defined with the following database access:

```
FINPLAN     R
CAPPLAN     W
PRODPLAN    N
```

He is assigned to Group Marketing, which has the following database access:

```
FINPLAN     N
CAPPLAN     N
PRODPLAN    W
```

His effective rights are set as:

```
FINPLAN     R
CAPPLAN     W
PRODPLAN    W
```

Example 2:

User Mary is defined with the following database access:

```
FINPLAN     R
PRODPLAN    N
```

She is assigned to Group Marketing, which has the following database access:

```
FINPLAN     N
PRODPLAN    W
```

Her effective rights are set as:

```
FINPLAN     R
PRODPLAN    W
```

In addition, Mary uses the filter artifact RED (for the database FINPLAN). The filter has two filter rows:

Access	Member Specification
Read	Actual
Write	Budget, @IDESCENDANTS(“New York”)

The Group Marketing also uses a filter artifact BLUE (for the database FINPLAN). The filter has two filter rows:

Access	Member Specification
Read	Actual, Sales
Write	Budget, Sales

Mary's effective rights from the overlapping filters, and the permissions assigned to her and her group:

R	Entire Finplan database
W	For all Budget data in the New York branch
W	For data values that relate to Budget and Sales

40 Security Examples in Native Security Mode

In This Chapter

The sample security problems and solutions described in this chapter are based on the Sample application and use security procedures described in Chapter 38, "User Management and Security."

Example 1: Users Require the Same Access to Databases

Three employees need to use Essbase—Sue Smith, Bill Brown, and Jane Jones. Each requires update access to all databases in the Sample application.

Solution When Using Essbase in Native Security Mode:

Because the users need update access to only one application, they do not need Administrator permission. Because the users do not need to create or delete applications, users, or groups, they need not be defined as special types of users with Create/Delete permission. These users need only Application Manager permission for the Sample application.

The Administrator should perform the following tasks:

1. Set up the users with Administration Services.

   See the *Oracle Essbase Administration Services Online Help*.

2. Create Sue, Bill, and Jane as ordinary users with Application Manager permission.

   If Sue, Bill, and Jane are created without Application Manager permission, assign Application Manager permission to the three users.

   See "Creating Users in Native Security Mode" on page 626 or "Granting Designer Permissions to Users and Groups in Native Security Mode" on page 630.

Example 2: Users Require Differing Access to Databases

Three employees need to use Essbase—Sue Smith, Bill Brown, and Jane Jones. Sue and Bill require full access to all databases in the Sample application. Jane requires full calculate access to all databases in the Sample application, but she does not need to define or maintain database definitions.

Solution When Using Essbase in Native Security Mode:

The Administrator should perform the following tasks:

1. Set up the users with Administration Services.

   See the *Oracle Essbase Administration Services Online Help*.

2. Create Sue and Bill as ordinary users with Application Manager permission.

   If Sue and Bill are created without Application Manager permission, assign them Application Manager permission.

   See "Creating Users in Native Security Mode" on page 626 or "Granting Designer Permissions to Users and Groups in Native Security Mode" on page 630.

3. Define global Calculate access for the Sample application as the Minimum Database Access setting to give all additional users Calculate access to all databases for the application.

   See "Setting Minimum Permissions for Applications" in the *Oracle Essbase Administration Services Online Help*.

4. Create Jane as an ordinary user with no additional permissions. She inherits Calculate access from the application global setting.

   See "Creating Users in Native Security Mode" on page 626.

Example 3: Users Require Differing Access to Databases; Users Will Be Added

Three employees need to use Essbase—Sue Smith, Bill Brown, and Jane Jones. Sue and Bill require full access to all databases in the Sample application. Jane requires full update and calculate access to all databases within the Sample application, but she will not define or maintain database definitions. Users will be added; all will require read access to all databases.

Solution When Using Essbase in Native Security Mode:

Because the current users have differing needs for application and database access, define their permissions individually. Then, to save time assigning individual read permissions for future users, make read the global setting for the application. (It does not matter in what order you assign the user permissions and the global access.)

The Administrator should perform the following tasks:

1. Set up the users with Administration Services.

   See the *Oracle Essbase Administration Services Online Help*.

2. Create or edit Sue and Bill as ordinary users with Application Manager permissions.

   See "Creating Users in Native Security Mode" on page 626 and "Granting Designer Permissions to Users and Groups in Native Security Mode" on page 630.

3. Create Jane as an ordinary user, and give her Calculate permission for the Sample application.

   See "Creating Users in Native Security Mode" on page 626 and "Granting Application and Database Access to Users and Groups in Native Security Mode" on page 629.

4. Define global read access for the Sample application as the minimum database access setting to give all additional users read access to all databases in the Sample application.

   See "Setting Minimum Permissions for Databases" in the *Oracle Essbase Administration Services Online Help*.

Example 4: Users Require Differing Access to Application and Databases

Three employees need to use Essbase—Sue Smith, Bill Brown, and Jane Jones. Sue requires full access only to the Sample application; Jane requires calculate access to all members of the Basic database; Bill requires read access to all members. No other users should have access to the databases.

Furthermore, Jane and Bill need to run report scripts that are defined by Sue.

Solution When Using Essbase in Native Security Mode:

Because the different users have different needs for application and database access, define the global access setting as None, and assign the user permissions individually.

The Administrator should perform the following tasks:

1. Set up the users with Administration Services. (Because Jane and Bill need to run the report scripts, they must use Administration Services.)

   See the *Oracle Essbase Administration Services Online Help*.

2. Create Sue as an ordinary user, and grant her Application Manager permission for the Sample application.

   See "Creating Users in Native Security Mode" on page 626 and "Granting Designer Permissions to Users and Groups in Native Security Mode" on page 630.

3. Create Jane as an ordinary user, and give her Calculate permission for the Sample application.

   See "Creating Users in Native Security Mode" on page 626 and "Granting Application and Database Access to Users and Groups in Native Security Mode" on page 629.

4. Create Bill as an ordinary user and give him read permission on the Sample application.

   See "Creating Users in Native Security Mode" on page 626 and "Granting Application and Database Access to Users and Groups in Native Security Mode" on page 629.

Example 5: Administrator Must Perform Maintenance

The Administrator, Sue Smith, needs to perform maintenance on the Sample application. She must make changes to the database outline and reload data. While she changes the application, Sue must prevent other users from connecting to the application.

Solution When Using Essbase in Native Security Mode:

Sue should perform the following tasks:

1. Disable the Allow Commands setting to prevent other users from connecting to the application and to prevent connected users from performing further operations.

   See "Clearing Applications of User Activity" in the *Oracle Essbase Administration Services Online Help*.

2. Check to see if any users have active locks.

   If any users have active locks, Sue's calculation or data load command might halt, waiting for access to the locked records. Sue can allow the users to complete their updates or clear their locks.

   See "Viewing Data Locks" in the *Oracle Essbase Administration Services Online Help*.

3. After confirming that no users have active locks, proceed to perform maintenance on the application.

Part VII

Enabling Multi-Language Applications Through Unicode

In Enabling Multi-Language Applications Through Unicode:

41 Understanding the Essbase Unicode Implementation

In This Chapter

The information in this chapter applies to block storage and aggregate storage databases.

About Unicode

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

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

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

Essbase conforms to version 2.1 of the Unicode Standard and uses UTF-8 encoding. See www.unicode.org.

Through its Unicode implementation, Essbase enables employees of global businesses to view, in their own languages, company information stored in Essbase databases. For example, using alias tables in their respective languages, users in Taiwan can view database reports in Chinese characters, and users in France can view the same reports in French characters.

The following topics describe the characteristics of the Essbase implementation of Unicode.

Note:

In Essbase, user-defined character sets (UDC) are not supported.

When to Use Unicode-Mode Applications

Consider working with Unicode-mode applications only in the following situations:

- You need to enable users with different languages to view, in their own languages and character sets, information from a common database.

  For example, using alias tables in Japanese and German, users in Japan and Germany can view information about a common product set in their own languages.

- You need to handle artifact names longer than non-Unicode-mode applications support.

  For example, application and database names need to include more than eight characters, or you are working with a multibyte character set, and you need to handle more characters in artifact names.

  See Appendix A, “Limits.”

- For a translated, multibyte Essbase implementation, you have experienced a “round-trip” problem, where two different bit values can map to the same character, which can occur in communications between multibyte operating systems and application programs.

  As Java applications, Administration Services and Provider Services always work in Unicode; therefore, no round-trip conversion errors occur.

When deciding whether to use Unicode-mode applications, consider the following points:

- Using non-Unicode text files with Unicode-mode applications requires an understanding of locales and care in managing them. Oracle recommends using UTF-8 encoded files to prevent errors that can cause database corruption.

  See “Managing File Encoding” on page 678.

- To work with Unicode-mode applications, custom client programs that were written to support non-Unicode-mode applications must be built to use the longer strings used by Unicode-mode applications. This task may be a simple rebuild or may involve reprogramming, depending on the design of the applications. Depending on how modified programs are coded, more memory may be required.

  See the *Oracle Essbase API Reference*.

Locales and the ESSLANG Variable

Essbase uses the ESSLANG variable to define the locale of a computer. For example, to support American English, you can set ESSLANG to `English_UnitedStates.Latin1@Binary`.

The ESSLANG variable includes a code-page specification that maps bit combinations to characters. Although a locale specification contains several kinds of information about the language and culture that the computer supports, Essbase uses only the code-page portion; cultural conventions and sort-order portions are not used.

For each Essbase Server installation, you must specify the ESSLANG variable, which should be set to the locale that is defined for the computer's operating system.

For client computers, specifying the ESSLANG variable is optional. If ESSLANG is defined, Essbase uses the ESSLANG value as the computer locale. If ESSLANG is not specified, the operating system locale is used.

The ESSLANG variable is set when you install Essbase. See the *Oracle Hyperion Enterprise Performance Management System Installation and Configuration Guide*.

For non-Unicode-mode clients and applications, the client and Essbase Server locale values must be the same. For Unicode-mode applications, client and Essbase Server locale values can be different.

Unicode and Non-Unicode Application Modes

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

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

Because Unicode-mode applications accept input files in non-Unicode-encoding and UTF-8, Essbase relies on locale indicators and user prompting to read or write non-Unicode-encoded files.

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

For Unicode-mode applications, the limits of most artifact names are longer than the limits in non-Unicode-mode applications, and the limits are calculated based on characters rather than bytes. See "Increased Name Lengths" on page 668.

Non-Unicode-mode applications support one character set that is defined by a locale value that must be the same for Essbase Server and all non-Unicode clients that work with non-Unicode-mode applications. By default, Essbase creates applications in non-Unicode mode.

You can define an application as Unicode-mode when you create the application (see "Creating Unicode-Mode Applications" on page 675), or you can migrate a non-Unicode-mode application to Unicode mode in a separate step (see "Migrating Applications to Unicode Mode" on page 675).

Note:

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

Unicode-mode and non-Unicode-mode applications can reside on the same Essbase Server.

Unicode and Non-Unicode Essbase Server Modes

Essbase Server is in Unicode mode when it has permission to create Unicode-mode applications and to migrate existing applications to Unicode mode. See "Setting Essbase Server to Unicode Mode" on page 674.

Whether Essbase Server is in Unicode mode affects only the creation and migration of applications. Regardless of the Essbase Server Unicode setting, you can work with both Unicode mode and non-Unicode-mode applications.

Increased Name Lengths

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

The maximum length of Unicode artifact names is based on the number of characters, regardless of how many bytes each character requires. The maximum length of non-Unicode artifact names is calculated in bytes.

Not limiting by bytes is advantageous for applications using multibyte character sets, such as Chinese and Japanese. For example, the limit for member names in Unicode-mode applications is 80 characters, even if they are multibyte characters, whereas the limit for member names in non-Unicode applications is 80 bytes. See Appendix A, "Limits."

Note:

The increased size limits may affect the size of the outline and user-written client programs.

To take advantage of longer name sizes, users may decide to work with Unicode-mode applications, even if all users work in the same locale (see "When to Use Unicode-Mode Applications" on page 666).

Compatibility Between Different Versions of Client Programs and Essbase Applications

Essbase supports Unicode-mode and non-Unicode-mode client programs.

Non-Unicode-mode client programs (for example, MaxL Shell and Spreadsheet Add-in):

- Communicate with Essbase API in short strings
- Cannot make changes to Unicode-mode outlines
- Cannot perform outline synchronization of Unicode-mode outlines

Unicode-mode client programs (for example, Administration Services Console and Smart View):

- Communicate with Essbase API in Unicode encoding, using long strings
- Cannot synchronize non-Unicode-mode outlines with multibyte characters

Note:

If you use Administration Services Console or another tool to create a MaxL script and save it in UTF-8 and then execute the script in MaxL Shell, MaxL Shell assumes the role of a Unicode-mode client. You can use this approach, for example, to update outlines through dimension builds. When you create the script, remember to include the UTF-8 signature. See "Encoding Indicators" on page 679.

Unicode-Enabled C API

Custom-written client programs used with pre-7.0 Essbase releases cannot be used with Unicode-mode applications because these custom programs use short strings and short buffers.

To provide restricted access to Unicode-mode applications, these older custom client programs, depending on how they are written, can be recompiled in a Unicode-enabled release of Essbase. When recompiled, the programs work with long buffers but short strings.

For complete access to Unicode-mode and non-Unicode-mode applications, existing custom applications must be modified using the Essbase API functions for Unicode. Rewritten and compiled clients work with long buffers and long strings for full Unicode support. See the *Oracle Essbase API Reference*.

Identification of Text Encoding and Locale

Essbase supports use of non-Unicode-encoded files, such as report and calculation scripts, with Unicode-mode applications. To identify a file's encoding type, Essbase looks for encoding indicators (UTF-8 signature or locale indicator). If a file does not contain either encoding indicator, and the file is not on Essbase Server, Administration Services prompts the user for the locale of the file. See "Encoding Indicators" on page 679.

The Essbase Unicode File Utility includes options to insert a UTF-8 signature or locale indicator in text files. Or you can use a text editor or other means to insert them. See "Essbase Unicode File Utility" on page 683.

Unicode-Enabled Administration Tools

To administer Unicode-mode applications, you can use Administration Services and MaxL Shell. With Administration Services Console, which is a Unicode-mode client, you can administer Unicode-mode and non-Unicode-mode applications.

Unicode-related administration activities include changing the Unicode-related mode of Essbase Server to enable or disable the following abilities:

- Creation of Unicode-mode applications
- Migration of non-Unicode-mode applications to Unicode mode
- Viewing of Unicode-related status of Essbase Server and applications

See Chapter 42, "Administering Unicode-Mode Applications".

Retrieval Tools

Essbase provides several methods for retrieving data from Unicode-mode and non-Unicode-mode applications.

Report Script

Report script output files are encoded to the encoding of the application. For example, if a report script is run against a database in a Unicode-mode application, the report script output is encoded in UTF-8; if run against a database in a non-Unicode-mode application, the output is encoded in the encoding of the application.

Spreadsheet

With Smart View, you can view data in Unicode-mode and non-Unicode-mode applications.

To run Smart View, you connect to Oracle Hyperion Provider Services. To install these programs, see the *Oracle Hyperion Enterprise Performance Management System Installation and Configuration Guide*.

For information about working with Smart View, see the *Oracle Hyperion Smart View for Office User's Guide*.

Note:

Spreadsheet Add-in does not support Unicode.

SQL Interface

With SQL Interface, you can load data from a Unicode-mode relational database to a Unicode-mode Essbase application. Table 66 shows supported encodings for Unicode-mode and non-Unicode-mode applications:

Table 66 SQL Interface Supported Encodings for Unicode-Mode and Non-Unicode-Mode Applications

Application mode	Relational Database Encoding
Non-Unicode	A supported non-Unicode encoding
Unicode	A supported non-Unicode encoding
Unicode	UTF-8

SQL Interface accepts user authentication (user name and password) and application information in UTF-8 encoding.

See "Supported Locales" on page 681.

Sample_U.Basic

To learn more about Unicode-mode applications, use the Sample_U.Basic application. Member names in Sample_U.Basic are in English.

Table 66 lists the non-English alias tables and their import files included in the Sample_U.Basic database:

Table 67 Non-English Alias Tables in the Sample_U.Basic Database

Language	Non-English Alias Table
Chinese	`nameschn.alt`
German	`namesger.alt`
Japanese	`namesjpn.alt`
Russian	`namesrsn.alt`

42 Administering Unicode-Mode Applications

In This Chapter

The information in this chapter applies to block storage and aggregate storage databases.

Setting Up a Computer for Unicode Support

Install the following tools (provided by third-party vendors) for working with UTF-8 encoded text on computers that manage Unicode-mode applications:

- UTF-8 fonts: For viewing UTF-8 encoded text that contains non-ASCII characters
- (Optional) Unicode editor (which includes the UTF-8 signature): For manual editing of data sources or other text files

Essbase provides the Essbase Unicode File Utility, which converts text files to UTF-8 encoding. See "Essbase Unicode File Utility" on page 683.

Defining Password Security

Essbase security is defined at the Essbase Server level. To create passwords, use characters encoded according to the code page specified in the ESSLANG variable on the Essbase Server computer.

In a multibyte character set environment, passwords cannot contain multibyte characters. Consider using the characters in the standard ASCII character set, in the range from 0 through 127, which includes the uppercase and lowercase letters in the standard English alphabet and the numerals 0 through 9. These characters are common to all code pages and to UTF-8.

Naming Applications and Databases

When you name applications and databases, use characters supported by the locale of the computer.

Managing Unicode-Mode Essbase Servers

The topics in this section discuss processes for managing Unicode-mode Essbase Servers.

Setting Essbase Server to Unicode Mode

By setting Essbase Server to Unicode mode, you grant permission for creating Unicode-mode applications and for migrating applications to Unicode mode.

➤ To set Essbase Server to Unicode mode, use a tool:

Tool	Topic	Location
Administration Services	Managing the Essbase Server Permission to Create Unicode-Mode Applications	*Oracle Essbase Administration Services Online Help*
MaxL	**alter system**	*Oracle Essbase Technical Reference*

Note:

You can work with Unicode-mode applications without Essbase Server being set to Unicode mode.

Viewing the Unicode-Related Mode of Essbase Server

The Unicode-related mode of Essbase Server indicates whether Essbase Server has permission to create Unicode-mode applications and to migrate applications to Unicode mode.

In Administration Services Console, the Unicode-related mode is shown as a server property.

In MaxL Shell, the Unicode-related mode is displayed as the configuration, with the following values:

- 2, for non-Unicode-mode applications
- 3, for Unicode-mode applications

➤ To see whether Essbase Server is set to Unicode mode, use a tool:

Tool	Topic	Location
Administration Services	Managing the Essbase Server Permission to Create Unicode-Mode Applications	*Oracle Essbase Administration Services Online Help*

Tool	Topic	Location
MaxL	**display system**	*Oracle Essbase Technical Reference*

Managing Unicode-Mode Applications

The topics in this section discuss processes for managing Unicode-mode applications.

Creating Unicode-Mode Applications

When you create an application, you can specify the application to be Unicode-mode-enabled.

➤ To create a Unicode-mode-enabled application, use a tool:

Tool	Topic	Location
Administration Services	Creating Unicode-Mode Applications	*Oracle Essbase Administration Services Online Help*
MaxL	**create application**	*Oracle Essbase Technical Reference*

Migrating Applications to Unicode Mode

When Essbase Server migrates a non-Unicode-mode application to Unicode mode, the character encoding in the application files is converted to UTF-8 encoding.

Before migrating application files to Unicode mode, perform the following tasks:

- Back up all application and database files in the application.

  See the *Oracle Hyperion Enterprise Performance Management System Backup and Recovery Guide*.

- If needed, apply the outline change files and then remove them.

  If present, outline change files reside in each database directory, with the database name as the file name and a `.chg` extension; for example, `/sample/basic/basic.chg`.

- To avoid mixing Unicode encoding and non-Unicode encoding in log files, back up log files and then clear or remove them.

  Log files end with any of the following extensions: `.log`, `.xcp`, and `.olg`.

- Grant Essbase Server permission to migrate applications to Unicode mode. See "Setting Essbase Server to Unicode Mode" on page 674.

➤ To migrate an application from non-Unicode mode to Unicode mode, use a tool:

Tool	Topic	Location
Administration Services	Migrating Applications to Unicode Mode	*Oracle Essbase Administration Services Online Help*
MaxL	**alter application**	*Oracle Essbase Technical Reference*

Text files, such as calculation scripts, report scripts, and data sources, are not converted to UTF-8 encoding. For Unicode-mode applications, text files can be encoded in UTF-8 or in non-Unicode locales. You can use Essbase Unicode File Utility to convert non-Unicode-encoded files to UTF-8. See "Essbase Unicode File Utility" on page 683.

Note:

If you choose to work with non-Unicode text files in your Unicode-mode application, ensure that you understand how Essbase determines the locales of non-Unicode text files. See "Managing File Encoding" on page 678.

Backing Up and Restoring Databases Between Unicode and Non-Unicode Applications

When working with Unicode and non-Unicode applications, Essbase does not allow a backed up database from a Unicode application to be restored to a non-Unicode application.

For a list of supported combinations of restoring backed up databases between Unicode and non-Unicode applications, see the *Oracle Hyperion Enterprise Performance Management System Backup and Recovery Guide*.

Viewing the Unicode-Related Mode of an Application

In Administration Services, the Unicode-related mode of an application is shown as an application property. In MaxL, it is displayed as the application type, with the following values:

- 2, for non-Unicode-mode applications
- 3, for Unicode-mode applications

➤ To see whether an application is a Unicode-mode application, use a tool:

Tool	Topic	Location
Administration Services	Monitoring Applications	*Oracle Essbase Administration Services Online Help*
MaxL	**display application**	*Oracle Essbase Technical Reference*

Using Control Characters in Report Scripts

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

Non-Unicode-mode applications that use single-byte code pages support the following Hexadecimal control characters in report scripts:

```
x20, x09, x0A, x0D, xB3, xC3, xC4, xC2, xC0, x0C, xB1
```

Unicode-mode applications and non-Unicode-mode applications that use multibyte code pages support the following Hexadecimal control characters in report scripts:

```
x20, x09, x0A, x0C, x0D
```

Working with Partitions

Consider the following situations when designing and working with partitions:

- Partitions cannot connect non-Unicode-mode databases to Unicode-mode databases.
- For transparent and replicated (but not linked) partitions, the application mode of both ends of the partitions must be the same—either Unicode mode or non-Unicode mode.
- All databases in a partition must be in the same encoding.
- Partitions across Unicode-mode databases can be administered by Unicode-mode clients; for example, Administration Services or MaxL scripts executed in Administration Services Console.
- Non-Unicode-mode outlines containing multibyte characters cannot be synchronized by Unicode-mode clients, such as Administration Services and MaxL scripts. You can, however, use a client that is not Unicode-enabled or a non-Unicode mode client, such as MaxL Shell (`essmsh`).

Working with Logs

By default, the Essbase agent log file is encoded to the locale specified by the ESSLANG variable defined for Essbase Server. You can use the UNICODEAGENTLOG configuration setting to write the agent log in UTF-8 encoding. With a UTF-8 font, all characters in the log file should be readable. See the *Oracle Essbase Technical Reference*.

Application and outline change log files for databases in Unicode-mode applications are UTF-8 encoded. To view:

- Application logs, use Log Viewer or Log Analyzer in Administration Services, or a UTF-8 editor
- Outline change logs, use a UTF-8 editor

You cannot change the encoding of log files related to Unicode-mode applications to non-Unicode encoding.

Managing File Encoding

Essbase supports many non-Unicode encodings (see "Supported Locales" on page 681). In addition, UTF-8 encoding is supported for Unicode-mode applications.

Because a bit combination can map to different characters in different encodings, you must understand how Essbase works with file encodings, particularly if you intend to use non-Unicode-encoded files with Unicode-mode applications.

Note:

Oracle recommends using UTF-8-encoded files for Unicode-mode applications shared across multiple locales. Using UTF-8 encoding is simpler, because you need not keep track of locales and encoding, and it can be more efficient, because Essbase Administration Server uses Unicode encoding internally, and no internal conversion between formats is needed.

The following topics describe how Essbase determines the encoding of files and how you manage files with different encoding.

How the Encoding of a File Is Determined

With non-Unicode-mode applications, Essbase and Administration Services assume that character text is encoded to the locale specified by the ESSLANG value defined for Essbase Server.

When Essbase works with Unicode-mode applications, it encodes character text in UTF-8 encoding internally and stores character text in UTF-8. Export files also are encoded in UTF-8. When Essbase works with Unicode-mode applications, it can also handle non-Unicode-encoded input files (such as script files, rules files, and data sources), converting them internally to UTF-8.

Note:

For Unicode-mode applications, Essbase requires that `*dbname*.cfg` query log settings file be encoded in UTF-8.

Essbase and Administration Services use file-encoding indicators (UTF-8 signature or locale indicator) to know whether a file is encoded in UTF-8 or in a supported non-Unicode encoding.

A locale indicator is optional for a non-Unicode input file whose encoding matches the locale of Essbase Server. If, however, the encoding does not match, you must provide a locale indicator. See "Encoding Indicators" on page 679.

UTF-8-encoded text files must include the UTF-8 signature.

Administration Services uses the following process to determine the encoding of a non-Unicode-encoded file:

1. If a locale indicator is present in the file, Administration Services uses the specified encoding.
2. If a locale indicator is not present, and the file is located within an application, Administration Services uses the encoding specified in the Essbase Server locale.
3. It a locale indicator is not present, and the file is not located within an application, Administration Services determines the encoding based on the type of file:
   - Text files: Administration Services prompts for the encoding.
   - Outline and rules files: Administration Services uses the encoding specified in the Essbase Server locale.

When Essbase performs a dimension build or data load, the rules file and data file can have different encodings. For example, the text in a rules file can be in UTF-8 encoding, and the data source can be encoded to a non-Unicode computer locale.

Note:

When you use Administration Services Console to create script files or data sources, the appropriate encoding indicator is automatically included in the file. If you use any other tool to create Unicode-encoded text files, you must ensure that the UTF-8 signature is included. Non-Unicode-encoded text files require a locale indicator if the encoding is different from the Essbase Server locale.

The following Essbase system text files are encoded to the locale specified by the ESSLANG value defined for Essbase Server:

- `essbase.cfg`
- ESSCMD scripts

Encoding Indicators

To properly interpret text, such as member names, Essbase must know how it is encoded. Many files contain an encoding indicator, but occasionally you may be prompted to specify the encoding; for example, in the following cases:

- Administration Services creates a file and stores it in a different location than the Essbase Server
- Administration Services reads a file created by a non-Unicode, pre-7.0 release of Essbase

The type of encoding indicator depends on the type of file:

- Files that are internal to applications and databases and that users cannot directly edit are primarily binary files and do not contain encoding indicators.

  Character text in these files is encoded based on the application mode:

  - Text in Unicode-mode application files is UTF-8 encoded.

    - Text in non-Unicode-mode application files is encoded to the locale specified in the ESSLANG variable of the Essbase Server where the application was created.
- Binary files that you can edit (such as outline and rules files)

  As needed, Essbase keeps track internally of whether the character text is in UTF-8 encoding. If not UTF-8, Essbase uses an internal locale indicator to identify the locale used for character text encoding.
- The following text files that you can edit use a UTF-8 signature or a locale indicator to indicate their encoding:
    - Calculation scripts
    - Report scripts
    - MaxL scripts
    - Data sources for dimension builds and data loads
    - Alias table import files (Administration Services requires alias table import files to be UTF-8 encoded)

Caution!

Do not use non-Unicode-encoded files containing locale indicators with pre-7.0 release Essbase Server installations, which are not Unicode enabled. To remove the locale indicators, use Essbase Unicode File Utility. See "Essbase Unicode File Utility" on page 683.

UTF-8 Signature

UTF-8-encoded text files must contain the UTF-8 signature, which is a mark at the beginning of a text file. This signature is visible in some third-party UTF-8 text editors.

Many UTF-8 text editors can create the UTF-8 signature, or you can use the Essbase Unicode File Utility to insert the UTF-8 signature into a file (see "Essbase Unicode File Utility" on page 683).

When you create a file using Administration Services Console, a UTF-8 signature is automatically inserted in the file.

Locale Indicator (Locale Header Record)

The locale indicator is a *locale header record*, which is an additional text record that identifies the encoding of a non-Unicode text file. You can insert the locale header record as the first record in a non-Unicode-encoded text file when you create the file, or you can use Essbase Unicode File Utility to add the header.

Caution!

Do not insert a locale header record in a UTF-8 encoded file. If a text file contains both types of encoding indicators, the locale header is read as the first data record.

The locale header record has the following format:

```
//ESS_LOCALE locale-name
```

locale-name is a supported Global C locale in the format that is used for the ESSLANG variable:

```
language_territory.code_page_name@sortsequence
```

The following example displays a locale header record for a specific Russian code page:

```
//ESS_LOCALE Russian_Russia.ISO-8859-5@Default
```

Note:

Essbase consults only the *code_page_name* portion of the record. The *sortsequence* specification does not affect sort sequences in report scripts.

The following rules also apply:

- After *locale-name*, use a blank, tab or <end of line> to end the header.
- You can include blanks or tabs in the following locations:
  - Before the keyword "`//ESS_LOCALE`"
  - Between "`//ESS_LOCALE`" and *locale-name*
  - After the locale specification

For compatibility, Administration Services Console saves calculation scripts on a pre-7.0 release Essbase Server installation in a different format. Instead of prefixing the locale with `//`, Administration Services Console inserts the locale header between calculation script comment indicators:

```
/* */
```

Supported Locales

The following supported locales can be used in locale header records, in Essbase Unicode File Utility, and as ESSLANG variable values:

```
Arabic_SaudiArabia.ISO-8859-6@Default
Arabic_SaudiArabia.MS1256@Default
Croatian_Croatia.ISO-8859-2@Croatian
Croatian_Croatia.MS1250@Croatian
CyrillicSerbian_Yugoslavia.ISO-8859-5@Default
CyrillicSerbian_Yugoslavia.MS1251@Default
Czech_CzechRepublic.ISO-8859-2@Czech
Czech_CzechRepublic.MS1250@Czech
Danish_Denmark.ISO-8859-15@Danish
Danish_Denmark.IBM500@Danish
Danish_Denmark.Latin1@Danish
Danish_Denmark.MS1252@Danish
Dutch_Netherlands.IBM037@Default
Dutch_Netherlands.IBM500@Default
Dutch_Netherlands.ISO-8859-15@Default
Dutch_Netherlands.Latin1@Default
Dutch_Netherlands.MS1252@Default
English_UnitedStates.IBM037@Binary
English_UnitedStates.IBM285@Binary
English_UnitedStates.IBM500@Binary
English_UnitedStates.MS1252@Binary
English_UnitedStates.Latin1@Binary
English_UnitedStates.US-ASCII@Binary
Finnish_Finland.IBM500@Finnish
Finnish_Finland.ISO-8859-15@Finnish
Finnish_Finland.Latin1@Finnish
Finnish_Finland.MS1252@Finnish
French_France.IBM297@Default
French_France.IBM500@Default
French_France.ISO-8859-15@Default
French_France.Latin1@Default
French_France.MS1252@Default
German_Germany.IBM273@Default
German_Germany.IBM500@Default
German_Germany.ISO-8859-15@Default
German_Germany.Latin1@Default
German_Germany.MS1252@Default
Greek_Greece.ISO-8859-7@Default
Greek_Greece.MS1253@Default
Hebrew_Israel.ISO-8859-8@Default
Hebrew_Israel.MS1255@Default
Hungarian_Hungary.ISO-8859-2@Hungarian
Hungarian_Hungary.MS1250@Hungarian
Italian_Italy.IBM280@Default
Italian_Italy.IBM500@Default
Italian_Italy.ISO-8859-15@Default
Italian_Italy.Latin1@Default
Italian_Italy.MS1252@Default
Japanese_Japan.IBM930@Binary
Japanese_Japan.JapanEUC@Binary
Japanese_Japan.MS932@Binary
Korean_Korea.MS1361@Binary
Korean_Korea.MS949@Binary
Norwegian_Norway.IBM500@Danish
Norwegian_Norway.ISO-8859-10@Danish
Norwegian_Norway.ISO-8859-15@Danish
Norwegian_Norway.ISO-8859-4@Danish
Norwegian_Norway.Latin1@Danish
Norwegian_Norway.MS1252.Default
Portuguese_Portugal.IBM037@Default
Portuguese_Portugal.IBM500@Default
Portuguese_Portugal.ISO-8859-15@Default
Portuguese_Portugal.Latin1@Default
Portuguese_Portugal.MS1252@Default
Romanian_Romania.ISO-8859-2@Romanian
Romanian_Romania.MS1250@Romanian
Russian_Russia.ISO-8859-5@Default
Russian_Russia.MS1251@Default
Serbian_Yugoslavia.ISO-8859-2@Default
Serbian_Yugoslavia.MS1250@Default
SimplifiedChinese_China.IBM935@Binary
SimplifiedChinese_China.MS936@Binary
SimplifiedChinese_China.UTF-8@Binary
```

```
Slovak_Slovakia.ISO-8859-2@Slovak
Slovak_Slovakia.MS1250@Slovak
Slovenian_Slovenia.ISO-8859-10@Slovenian
Slovenian_Slovenia.ISO-8859-2@Slovenian
Slovenian_Slovenia.ISO-8859-4@Slovenian
Slovenian_Slovenia.MS1250@Slovenian
Spanish_Spain.IBM500@Spanish
Spanish_Spain.ISO-8859-15@Spanish
Spanish_Spain.Latin1@Spanish
Spanish_Spain.MS1252@Spanish
Swedish_Sweden.IBM500@Swedish
Swedish_Sweden.ISO-8859-15@Swedish
Swedish_Sweden.Latin1@Swedish
Swedish_Sweden.MS1252@Swedish
Thai_Thailand.MS874@Thai
TraditionalChinese_Taiwan.EUC-TW@Binary
TraditionalChinese_Taiwan.IBM937@Binary
TraditionalChinese_Taiwan.MS950@Binary
Turkish_Turkey.ISO-8859-3@Turkish
Turkish_Turkey.ISO-8859-9@Turkish
Turkish_Turkey.MS1254@Turkish
Ukrainian_Ukraine.ISO-8859-5@Ukrainian
Ukrainian_Ukraine.MS1251@Ukrainian
```

Essbase Unicode File Utility

You can use Essbase Unicode File Utility to convert non-Unicode-encoded text files to UTF-8 encoding, or to add or delete encoding indicators. This program supports the following operations:

- Converting non-Unicode-encoded text files to UTF-8 encoding, including the UTF-8 signature.
- Adding UTF-8 signatures to UTF-8-encoded text files.
- Inserting a locale header record at the beginning of non-Unicode-encoded text files.
- Adding a locale indicator to outline files (`.otl`) or rules files (`.rul`).
- Removing locale indicators from files (the utility cannot remove UTF-8 signatures).

Located in the `*ESSBASEPATH*/bin` directory, Essbase Unicode File Utility is called `essutf8.exe` (in Windows) or `ESSUTF8` (in UNIX). You can use the utility program with the following files:

- Calculation scripts
- Report scripts
- MaxL scripts
- Text data sources for dimension builds and data loads
- Alias table import files
- Outline files
- Rules files

For information about this utility and its command syntax, see the *Oracle Essbase Technical Reference*.

Part VIII

Maintaining Essbase

In Maintaining Essbase:

43 Running Essbase Servers, Applications, and Databases

In This Chapter

Essbase Executable Files

Table 68 lists the Essbase server and client executable files:

Table 68 Main Essbase Executable Files

Executable File*	Description	Location	See
`essbase.exe`	Essbase Server Agent process	*ESSBASEPATH*/bin	"Understanding the Agent" on page 688
`esssvr.exe`	Application server process	*ESSBASEPATH*/bin	"Starting and Stopping Applications" on page 701
`essmsh.exe`	MaxL Shell	*ESSBASEPATH*/bin	*Oracle Essbase Technical Reference*
`esscmd.exe`	ESSCMD command-line client interface	*ESSBASEPATH*/bin	*Oracle Essbase Technical Reference*
`adminsvr.exe` or `startEAS.exe`	Essbase Administration Server executable	*HYPERION_HOME*/products/Essbase/eas/server/bin	*Oracle Essbase Administration Services Online Help*

Executable File*	Description	Location	See
`admincon.exe`	Administration Services Console application	*HYPERION_HOME*/ products/ Essbase/eas/ server/bin	*Oracle Essbase Administration Services Online Help*

*On UNIX, files do not have the `.exe` extension.

Understanding the Agent

Launching the Agent executable file, `essbase.exe`, starts the Essbase Server Agent process. The Agent process starts and stops all applications and acts as the traffic coordinator for Essbase Server.

On the computer where Essbase Server is installed, the Agent is accessible only from the server console, which is the primary terminal, or monitor, connected to the server computer.

When you start Essbase Server in the foreground, the Agent becomes active in an operating system window. In the Agent window, you can view release and license information, enter login and administrative commands, and monitor the behavior of Essbase Server. On Windows, Essbase can be accessed only from the server console. On UNIX, a telnet session is used to access Essbase remotely.

When you start Essbase Server in the background, the terminal becomes free for other input, and the Agent activities are not visible in the terminal. See "Starting Essbase Server as a Background Process" on page 691.

The agent log is called the Essbase Server log. See "Viewing the Essbase Server and Application Logs" on page 742.

Multithreading

`essbase.exe` and `esssvr.exe` (`ESSBASE` and `ESSSVR` on UNIX) are multithreading and symmetric multiprocessing (SMP) applications. Multithreading ensures high performance in a client-server environment. SMP provides scalability when a single server computer hosts multiple Essbase applications. Essbase Server uses POSIX kernel threads, which are included in UNIX operating systems.

By default, the number of threads is based on the number of licensed ports, as shown in Table 69. The number of ports represents the number of concurrent connections that Essbase supports. Essbase provides one reserve port for the system administrator, which is used to log off users when all other ports are in use.

Table 69 Licensed Ports and Multithreading

Number of Licensed Ports	Default Number of Threads
1–5 ports	5
6–10 ports	10

Number of Licensed Ports	Default Number of Threads
11+ ports	20

You can set the number of threads for the Agent or Essbase Server in the `essbase.cfg` file using the AGENTTHREADS, AGTSVRCONNECTIONS, and SERVERTHREADS configuration settings. See the *Oracle Essbase Technical Reference*.

Note:

Enabling hyperthreading on the computer on which Essbase Server runs is not recommended.

List of Agent Commands and Equivalents

➤ To display a list of available Agent commands, press Enter in the operating system window where you started Essbase Server in the foreground.

Table 70 describes each Agent command, and the MaxL, ESSCMD, or Administration Services equivalents:

Table 70 Agent Commands and MaxL, ESSCMD, or Administration Services Equivalents

Agent Command	Function	MaxL, ESSCMD, or Administration Services Equivalent
START *appname*	Starts the specified application.	● MaxL: **alter system load application *appname***; ● ESSCMD: LOADAPP ● Administration Services: Start, then Application on the application node in Enterprise View
STOP *appname*	Stops the specified application.	● MaxL: **alter system unload application *appname***; ● ESSCMD: UNLOADAPP ● Administration Services: Stop, then Application on application node in Enterprise View
USERS	Displays a list of users connected to the Essbase Server. The following information is displayed: ● Names of users connected to the Essbase Server ● Number of ports installed ● Number of connections ● Application to which each user is connected ● Database to which each user is connected	● MaxL: **display user;** (lists all users and shows which users are logged on) ● ESSCMD: LISTUSERS (lists all users) ● Administration Services: Edit, then Sessions on the server node in Enterprise View
PORTS	Displays the number of ports installed on the Essbase Server and the number of ports in use.	● MaxL: **display system;** (to display available unused ports) ● ESSCMD: N/A ● Administration Services: Edit, then Properties (Essbase Server Properties window, Statistics tab) on the server node in Enterprise View

Agent Command	**Function**	**MaxL, ESSCMD, or Administration Services Equivalent**
LOGOUTUSER *user*	Disconnects a user from the Essbase Server and frees a port. This command requires the Essbase system password.	● MaxL: **alter system logout session by user *username***; ● ESSCMD: LOGOUTUSER ● Administration Services: Edit, then Sessions on the server node in Enterprise View
PASSWORD	Changes the system password that is required to start the Essbase Server. This command requires the Essbase system password.	● MaxL: **alter user *`system_administrator`* set password *`password`***; ● ESSCMD: SETPASSWORD ● Administration Services: N/A
COMPACT	Enables compaction of the security file when the Agent is running. See "Managing Security-File Fragmentation" on page 645. **Note:** Essbase compacts the security file automatically each time the Agent is stopped.	● MaxL: **alter system compact security file;** ● ESSCMD: N/A ● Administration Services: Compact security file on the server's security node in Enterprise View
DUMP *filename*	Dumps information from the Essbase security file (`essbase.sec`) to a specified file in text (ASCII) format. If you do not supply a path with the filename, the file is saved to the *`ESSBASEPATH`*`/bin` directory. This command requires the Essbase system password. **Note:** You cannot use the DUMP command against an Essbase Server that is run as a service. If the server is running as a service, use the Export Security File command in Administration Services or the **export security_file** MaxL statement.	● MaxL: **export security_file;** ● ESSCMD: N/A ● Administration Services: Export Security File on the server's security node in Enterprise View See "Exporting the Security File" on page 646. **Note:** You can use the export security file command against an Essbase Server that is run as a service.
VERSION	Displays the Essbase Server software version number.	● MaxL: **display system;** ● ESSCMD: GETVERSION ● Administration Services: Edit, then Properties (Essbase Server Properties window, License tab) on the server node in Enterprise View
HELP	Lists all valid Agent commands and their respective functions. Same as pressing Enter.	N/A
QUIT and EXIT	Shuts down all open applications and stops Essbase Server.	● MaxL: **alter system shutdown;** ● ESSCMD: SHUTDOWNSERVER ● Administration Services: Stop on the server node in Enterprise View

Starting and Stopping Essbase Server

To start Essbase Server, you must have Administrator permissions.

Note:

You cannot start Essbase Server from ESSCMD or MaxL.

Starting Essbase Server in the Foreground

Starting Essbase Server in the foreground starts the Agent in an operating system window, in which you can enter commands and monitor Essbase Server activities.

When starting an Essbase Server for the first time (for example, after installing Essbase), the administrator enters a company name, Essbase Server system administrator user ID, and a system password. This information is saved in the Essbase security file (`essbase.sec`). As long as `essbase.sec` is present in your Essbase installation, you are not prompted to provide this information when subsequently starting Essbase Server.

To start Essbase Server in the foreground, use a method for the operating system on which Essbase Server runs (these examples assume that Essbase Server had been started at least once before):

- For UNIX, enter the following command at the operating system prompt:

  ```
  ESSBASE
  ```
- For Windows, choose a method:
  - Enter the following command at the operating system prompt:

    ```
    essbase
    ```
  - On the Start menu, select Programs, then Oracle EPM System, then Essbase, then Essbase Server, and then Essbase.
  - In the file system, double-click the `essbase.exe` file.
  - On the Start menu, select Run and enter `essbase`.

Starting Essbase Server as a Background Process

A system administrator may run Essbase Server as a background process when working in batch mode (for example, when using a UNIX shell script or Windows `.bat` file to run multiple tasks, such as starting and logging onto Essbase Server, loading data, and running calculation scripts and report).

Also, on Windows, running Essbase Server as a background process allows a system administrator to use Windows settings to improve performance of applications running in the foreground.

If you start Essbase Server in the background, these conditions apply:

- You do not have access to Agent commands.
- You cannot shut down Essbase Server from the Agent. You must use MaxL or ESSCMD.
- You cannot access the application server window to monitor a running application. You must access this information from the application log (`*ARBORPATH*/app/*appname*/*appname*.log`).

- You cannot monitor Essbase Server activity using the Agent. You must access this information from the Essbase Server log (`*HYPERION_HOME*/logs/essbase/essbase.log`).

You can run instances of the Agent as Windows services. You must first use EPM System Configurator to register the service. See the *Oracle Hyperion Enterprise Performance Management System Installation and Configuration Guide*.

To start Essbase Server in the background on UNIX, or on Windows systems utilizing a UNIX-like shell such as MKS, enter the following command at a command prompt:

```
essbase -b &
```

Using the ampersand (&) at the end of the command is optional; however, if you do not use "&," the command prompt is not returned after Essbase Server is started.

Note:

On Windows, unless you are using a UNIX-like shell such as MKS, the ampersand (&) has no effect. Essbase Server starts in the background, but control of the command prompt is not returned. You may need to press the Enter key twice before the command prompt returns.

On UNIX systems, to find out if Essbase Server is already running in the background, enter the following command at a command prompt:

```
ps -ef | grep ESS
```

If Essbase Server is running in the background, it appears in the process list.

Hiding Essbase Server Passwords on HP-UX and Solaris

On HP-UX and Solaris, the `ps -ef` utility creates a process listing that includes the system password.

Note:

On IBM AIX, the Essbase Server system password is hidden automatically.

➤ To hide the Essbase Server system password:

1 **Create a shell script, named `essbase.secure`, that contains these commands:**

```
#!/bin/sh
PASS=$1
ESSBASE -b -secure << EOF &
${PASS}
EOF
```

2 **To launch the Agent, use this command:**

```
essbase.secure password
```

- To have the script to return to a command prompt without manually entering a carriage return or to imbed the script within a larger script, execute the script with this command:

```
essbase.secure &
```

- (Optional) To redirect the standard output to a file named `nohup.out`, which is useful when running the script in a nonactive terminal session, use this command:

```
nohup essbase.secure &
```

Changing the Essbase Server System Password

You can change the password that is required to start Essbase Server.

Note:

Changing the system password does not change the connection password for the Essbase system Administrator.

➤ To change the Essbase Server system password, use a tool:

Tool	Topic	Location
Agent	password	Enter the Agent command at the command prompt in the Essbase Server console window. Enter the current system password. Enter the new system password; then re-enter it.
MaxL	**alter user *system_administrator* set password *password***	*Oracle Essbase Technical Reference*
ESSCMD	SETPASSWORD	*Oracle Essbase Technical Reference*

Essbase verifies that the system password has been updated.

Stopping Essbase Server

You need Administrator permissions to stop or shut down Essbase Server.

➤ To stop Essbase Server and all running applications, use a tool:

Tool	Topic	Location
Agent	quit exit	Enter the Agent command at the command prompt in the Essbase Server console window.
Administration Services	Stopping Essbase Server	*Oracle Essbase Administration Services Online Help*
MaxL	**alter system shutdown**	*Oracle Essbase Technical Reference*

Tool	Topic	Location
ESSCMD	SHUTDOWNSERVER	*Oracle Essbase Technical Reference*

If you stop the Agent by closing the Agent window or by pressing Ctrl + C, the next time you start the database, Essbase rolls back any transactions that were in progress. See "Rollback with Committed Access" on page 783 or "Rollback with Uncommitted Access" on page 785.

Caution!

When running Essbase Server as a Windows service, do not stop the service from Windows Control Panel. Doing so is comparable to issuing a kill command on UNIX platforms and may cause data corruption.

Starting Essbase Server Remotely from Administration Services Console

You can start Essbase Server remotely from the Enterprise View tree in Administration Services Console. To enable this functionality, you configure and start Remote Start Server on the Essbase Server computer. Remote Start Server then handles requests from Administration Services to start Essbase Server.

Each Essbase Server installation includes all files necessary to configure and start Remote Start Server. To start different Essbase Servers from Administration Services Console, you must configure and start a Remote Start Server instance for each Essbase Server.

See these topics:

Configuring Remote Start Server

Before starting Remote Start Server, you must configure it if any of the following conditions apply:

- Essbase Server is configured to run on a non-default port.
- Environment variables must be set when Essbase Server is started.
- The default port used for Remote Start Server (9010) is being used by another program.
- More than one instance of Essbase Server is installed on one computer.

If none of these conditions apply, you are ready to start Remote Start Server. Skip to "Starting and Stopping Remote Start Server" on page 699.

To configure Remote Start Server, see the following topics.

About the Server.Properties Configuration File

Each Essbase Server installation contains a configuration file, `server.properties`, in the *`ESSBASEPATH`*`/bin` directory, which you can use to configure Remote Start Server.

The default configuration file contains the following lines:

```
Server1=localhost
localhostExecutable=PathToEssbase.exe
```

Where *`PathToEssbase.exe`* is the location of the Essbase Server executable file.

When editing the configuration file, keep in mind the following information:

- You can edit the file in any text editor.
- On Windows, you must escape the backslash (\) with an additional backslash (\\), as shown in the following examples.
- The structure of each line in the file:

  `key=keyValue`
- The value for the `Server1` key may be any one of the following:
  - `localhost`
  - *`machineName`*
  - *`machineName`*`:`*`port`*
- If you change the value of `Server1` to *`machineName`*, you must use *`machineName`* in place of `localhost` as the key prefix for any rows you add to the file. For example, on UNIX:

  ```
  Server1=jdoe2
  jdoe2Executable=Hyperion/products/Essbase/EssbaseServer/bin/essbase.exe
  ```
- If you change the value of `Server1` to *`machineName`*:*`port`*, you must use `Server1` in place of `localhost` as the key prefix for any rows you add to the file. For example, on Windows:

  ```
  Server1=jdoe2:4050
  Server1Executable=C:\\Hyperion\\products\\Essbase\\EssbaseServer\\bin\
  \essbase.exe
  ```
- When adding environment rows to the file, you first specify how many environment rows you are adding by using the `localhostEnvRows` setting. For example:

  ```
  localhostEnvRows=3
  localhostEnv1=Variable=VariableValue
  localhostEnv2=Variable=VariableValue
  localhostEnv3=Variable=VariableValue
  ```

  See "Setting Environment Variables for Startup" on page 696.
- You can set a non-default port for Remote Start Server to use.

The following sample configuration file for Windows sets the Essbase Server port to 4050; sets the *`ARBORPATH`* environment variable; and sets Remote Start Server port to 9030:

```
Server1=jdoe2:4050
Server1EnvRows=1
Server1Env1=ARBORPATH=C:\\Hyperion\\products\\Essbase\\EssbaseServer
Server1Executable=C:\\Hyperion\\products\\Essbase\\EssbaseServer\\bin\
```

```
\essbase.exe
ServerPort=9030
```

Specifying a Nondefault Essbase Server Port

In the configuration file for Remote Start Server, you must specify the port number being used for Essbase Server if Essbase Server is configured to run on a non-default port.

If you are using a non-default port because you are using multiple instances of Essbase Server on one computer, see "Configuring Remote Start Server for Multiple Instances of Essbase Server" on page 698.

➤ To specify a non-default Essbase Server port:

1 **On the Essbase Server computer, open the `server.properties` configuration file. For example:**

```
ESSBASEPATH/bin/server.properties
```

2 **Change the following lines from:**

```
Server1=localhost
localhostExecutable=PathToEssbase.exe
```

to

```
Server1=MachineName:Port
Server1Executable=PathToEssbase.exe
```

For example, on Windows:

```
Server1=jdoe2:4050
Server1Executable=C:\\Hyperion\\products\\Essbase\\EssbaseServer\\bin\
\essbase.exe
```

Setting Environment Variables for Startup

If environment variables must be set when starting Essbase Server remotely, you must add environment information to the Remote Start Server configuration file.

- On Windows, if you are not using multiple instances of Essbase Server on the same computer, you need not set environment variables. See "Configuring Remote Start Server for Multiple Instances of Essbase Server" on page 698.
- On UNIX, the environment must be set properly when starting Essbase Server. If the environment is not set, you must specify environment variable information in the Remote Start Server configuration file to be used for startup. Each UNIX platform has different environment variable requirements; see the *Oracle Hyperion Enterprise Performance Management System Installation and Configuration Guide*.

➤ To set environment variables for startup:

1 **From the Essbase Server computer, open the `server.properties` configuration file. For example:**

```
ESSBASEPATH/bin/server.properties
```

2 **Add lines for each environment variable that you must specify, using the format described in "About the Server.Properties Configuration File" on page 695.**

The number of environment rows that must be set depends on your platform.

The following sections show examples of how to set environment variables for different UNIX platforms, using the default Essbase Server port and an English ESSLANG value.

Solaris Example:

```
Server1=localhost
localhostEnvRows=4
localhostEnv1=ARBORPATH=/vol1/essbase
localhostEnv2=LD_LIBRARY_PATH=/vol1/common/ODBC/Merant/
  5.2/lib:/vol1/essbase/bin;$LD_LIBARY_PATH
localhostEnv3=ESS_JVM_OPTION1=-Xusealtsigs
localhostEnv4=ESSLANG=English_UnitedStates.Latin1@Binary
localhostExecutable=/vol1/essbase/bin/ESSBASE
```

AIX Example:

```
Server1=localhost
localhostEnvRows=3
localhostEnv1=ARBORPATH=/vol1/essbase
localhostEnv2=LIBPATH=/vol1/common/ODBC/Merant/
  5.2/lib:/vol1/essbase/bin;$LIBPATH
localhostEnv3=ESSLANG=English_UnitedStates.Latin1@Binary
localhostExecutable=/vol1/essbase/bin/ESSBASE
```

HP-UX Example

```
Server1=localhost
localhostEnvRows=3
localhostEnv1=ARBORPATH=/vol1/essbase
localhostEnv2=SHLIB_PATH=/vol1/common/ODBC/Merant/
  5.2/lib:/vol1/essbase/bin;$SHLIB_PATH
localhostEnv3=ESSLANG=English_UnitedStates.Latin1@Binary
localhostExecutable=/vol1/essbase/bin/ESSBASE
```

Specifying a Nondefault Remote Start Server Port

By default, Remote Start Server is configured to run on port 9010. If this port is being used by another program, you must specify a different port number before you start Remote Start Server. The port number must be specified in the Remote Start Server configuration file on the Essbase Server computer and in a configuration file on the Essbase Administration Server computer.

➤ To specify a nondefault Remote Start Server port:

1 **On the Essbase Server computer, open the configuration file. For example:**

ESSBASEPATH/bin/server.properties

2 **Add the following line:**

ServerPort=*PortNumber*

For example:

```
ServerPort=9030
```

3 **On the Essbase Administration Server computer, open the following configuration file:**

```
EASPATH/server\olapadmin.properties
```

4 **Add one of the following lines:**

- If you want Remote Start Server to use a specific port for a specific Essbase Server, add the following line:

  ```
  EssbaseServerName.REMOTE_START_SERVER=PortNumber
  ```

  For example:

  ```
  jdoe2.REMOTE_START_SERVER=9030
  ```

- If you want Remote Start Server to use the same, non-default port for all Essbase Servers, add the following line:

  ```
  REMOTE_START_SERVER=PortNumber
  ```

  For example:

  ```
  REMOTE_START_SERVER=9030
  ```

Configuring Remote Start Server for Multiple Instances of Essbase Server

If multiple instances of Essbase Server are installed on one computer, you must configure and start Remote Start Server for only one of the installations.

Note:

For information about installing multiple Essbase Server instances, see the *Oracle Hyperion Enterprise Performance Management System Installation and Configuration Guide*.

➤ To configure Remote Start Server for multiple instances of Essbase Server on one computer:

1 **Select one of the Essbase Server instances and open its configuration file. For example:**

```
ESSBASEPATH/bin/server.properties
```

2 **Add the following lines to the file:**

```
Server2=MachineName:PortNumber
Server2EnvRows=1
Server2Env1=ARBORPATH=ARBORPATHvalue
Server2Executable=PathToEssbase.exe
```

For example, on Windows:

```
Server2=jdoe2:4050
Server2EnvRows=1
Server2Env1=ARBORPATH=C:\\Hyperion\\products\\Essbase\\EssbaseServer-2
Server2Executable=C:\\Hyperion\\products\\Essbase\\EssbaseServer-2\\bin\
\essbase.exe
```

Note that *ARBORPATH* must be set explicitly for the second server instance. Also, for UNIX platforms, if the environment is not already set for either Essbase Server instance, you must specify environment variable information in the Remote Start Server configuration file. For information about the environment variables that are required for each UNIX platform, see the *Oracle Hyperion Enterprise Performance Management System Installation and Configuration Guide*.

The following sections show examples for Windows and UNIX (Solaris).

Windows Example

The default configuration for `localhost` and for the second instance of Essbase Server (`jdoe2`), which runs on port 4050.

```
Server1=localhost
localhostExecutable=C:\\Hyperion\\products\\Essbase\\EssbaseServer\\bin\
\essbase.exe
Server2=jdoe2:4050
Server2EnvRows=1
Server2Env1=ARBORPATH=C:\\Hyperion\\products\\Essbase\\EssbaseServer-2
Server2Executable=C:\\Hyperion\\products\\Essbase\\EssbaseServer-2\\bin\
\essbase.exe
```

UNIX Example (Solaris)

The configuration for `localhost` and for the second instance of Essbase Server (`jdoe2`), which runs on port 4050. The environment is set for both Essbase Server instances.

```
Server1=localhost
localhostEnvRows=4
localhostEnv1=ARBORPATH=/vol1/essbase
localhostEnv2=LD_LIBRARY_PATH=/vol1/common/ODBC/Merant/
  5.2/lib:/vol1/essbase/bin;$LD_LIBARY_PATH
localhostEnv3=ESS_JVM_OPTION1=-Xusealtsigs
localhostEnv4=ESSLANG=English_UnitedStates.Latin1@Binary
localhostExecutable=/vol1/essbase/bin/ESSBASE

Server2=jdoe2:4050
Server2EnvRows=4
Server2Env1=ARBORPATH=/vol2/essbase-2
Server2Env2=LD_LIBRARY_PATH=/vol2/common/ODBC/Merant/
  5.2/lib:/vol2/essbase-2/bin:$LD_LIBRARY_PATH
Server2Env3=ESS_JVM_OPTION1=-Xusealtsigs
Server2Env4=ESSLANG=English_UnitedStates.Latin1@Binary
Server2Executable=/vol2/essbase-2/bin/ESSBASE
```

Starting and Stopping Remote Start Server

After Remote Start Server is started, you can start Essbase Server.

Note:

On UNIX platforms, Remote Start Server can start an Essbase Server only if Essbase Server is installed by the same user name that was used to start Remote Start Server. If Remote Start Server is started by the root user, it can start an Essbase Server that was installed by any user.

➤ To start Remote Start Server:

- On Windows, launch:

  ESSBASEPATH/bin/remoteStart.exe
- On UNIX, launch:

  ESSBASEPATH/bin/remotesvr

➤ To stop Remote Start Server:

- On Windows platforms, launch

  ESSBASEPATH/bin/stopsvr.exe
- On UNIX platforms, launch

  ESSBASEPATH/bin/stopsvr

Running Remote Start Server as a Windows Service

You can install and start Remote Start Server as a Windows service.

To install and start Remote Start Server as a Windows service, launch:

ESSBASEPATH/bin/install_service.bat

`install_service.bat` installs Remote Start Server as a Windows service (named Essbase Remote Start Server) and starts the service. You can then manage the service from Windows Control Panel or by using `net start` and `net stop` commands.

To remove the Essbase Remote Start Server Windows service, launch:

ESSBASEPATH/bin/remove_service.bat

Starting Essbase Server from Administration Services Console

After starting Remote Start Server, you can start Essbase Server from Enterprise View in Administration Services Console.

Before you start Administration Services, ensure that the Essbase Servers you want to manage are started. First start Essbase Administration Server, and then start Administration Services Console.

To start Administration Services, see the *Oracle Essbase Administration Services Online Help*.

➤ To start Essbase Server from Enterprise View:

1. **From Enterprise View, select the Essbase Server node that you want to start.**
2. **Right-click and select Start.**
3. **When prompted, enter the password for Essbase Server.**

The request is sent to Remote Start Server, which then starts Essbase Server. If Remote Start Server is not running, an error message is displayed in Administration Services Console.

Note:

Essbase Server starts in the background on all platforms, with the password hidden.

Starting and Stopping Applications

When an application is started, Essbase loads it and all associated databases into memory on the Essbase Server computer. All client requests for data, such as data loads, calculations, reports, and spreadsheet lock and sends, are then handled through the application server process (`ESSSVR`). The application server is always started by the Agent.

Multiple application servers can run on Essbase Server concurrently. On Windows, a separate window opens for each ESSSVR process that is running. If an application contains multiple running databases, all databases are managed by the one application server.

When you stop an application, Essbase unloads all information and databases from memory on the Essbase Server computer and closes the application server process.

Starting an Application

When you start an application, the following actions can happen:

- Users can connect to the application.
- The application can respond to commands from the Agent.
- Users can change the settings of the application.
- Data and user security are enabled.
- Each database in the application can start.

➤ To start an application, use a tool:

Tool	Topic	Location
Agent	START *appname*	Enter the Agent command at the command prompt in the Essbase Server console window.
Administration Services	Starting Applications	*Oracle Essbase Administration Services Online Help*
MaxL	**alter system load application**	*Oracle Essbase Technical Reference*
ESSCMD	LOADAPP or SELECT	*Oracle Essbase Technical Reference*

The application starts and, if you are running on Windows, opens the application server window on the Essbase Server computer.

You can also start an application by completing any of these actions:

- Starting a database within an application. See "Starting a Database" on page 703.
- Saving an outline to Essbase Server. (Opening an outline does not start an application.)

You can set options that control how applications start:

- startup (Allow user to start application): If an application is stopped, and a user attempts to retrieve data from any databases within that application, the application starts on the Essbase Server computer automatically.
- autostartup (Start application when Essbase starts): Users may experience better initial performance when they make requests of databases in that application, because the application and databases are already loaded into memory on the Essbase Server computer.

➤ To control how applications are started, use a tool:

Tool	Topic	Location
Administration Services	Configuring Applications to Start Automatically	*Oracle Essbase Administration Services Online Help*
MaxL	**alter application enable startup** **alter application enable autostartup**	*Oracle Essbase Technical Reference*

Stopping an Application

Stop applications properly to prevent the databases within them from becoming corrupt. When you stop an application, transactions may be running. If you stop an application using any of the proper methods (see the following table), the application does not stop if a calculation or data load is in progress. Instead, Essbase displays a message in the Agent console.

If you stop the Agent by closing the server console window or by pressing Ctrl+C, the application stops, and the next time you start the application, Essbase rolls back any transactions that were in progress. See "Rollback with Committed Access" on page 783 or "Rollback with Uncommitted Access" on page 785.

➤ To properly stop applications, use a tool:

Tool	Topic	Location
Agent	stop *appname*	Enter the Agent command at the command prompt in the Essbase Server console window.
Administration Services	Stopping Applications	*Oracle Essbase Administration Services Online Help*
MaxL	**alter system unload application**	*Oracle Essbase Technical Reference*
ESSCMD	UNLOADAPP	*Oracle Essbase Technical Reference*

Stopping an Application Improperly

Sometimes, stopping the application server process improperly, by closing the application server window, is necessary; for example, if the application server is corrupted and is not processing client requests.

➤ To stop the application improperly, use a method for the operating system on which Essbase Server runs:

- For UNIX platforms, kill the `ESSSVR` process.

  You can use the ps output to identify individual applications. If an application freezes, you can stop the application by using this command:

  ```
  kill -9 <pid>
  ```

- For Windows, choose a method:
  - Perform a Windows operating system End Task.

    Windows does not display process IDs for individual Essbase applications—all of the running Essbase applications are displayed as undifferentiated `ESSSVR` processes, preventing you from stopping a single application in the event that the application freezes.
  - Click the Close button in the upper-right corner of the application server window.
  - Taskkill the process ID.

    You can find the process ID for individual application servers in the *`HYPERION_HOME`*`/logs/essbase/essbase.log` file. When the server starts, a line like the following is displayed in the Essbase Server log:

    ```
    Application [Sample] started with process id [225]
    ```

Starting and Stopping Databases

Starting a database loads the database into memory on the Essbase Server computer. Stopping a database unloads all database information from memory.

Starting a Database

When Essbase starts a database and loads it to memory, the entire index cache for that database is allocated in memory automatically. The data cache and data file cache are allocated as blocks requested from Essbase clients.

When you start an application, Essbase loads the application and its databases into memory on the Essbase Server computer. When you start a database from an application that is not started, the application is loaded into memory along with all its related databases.

➤ To start a database, use a tool:

Tool	Topic	Location
Agent	START *appname*	Enter the Agent command at the command prompt in the Essbase Server console window.
Administration Services	Starting Databases	*Oracle Essbase Administration Services Online Help*
MaxL	**alter application load database**	*Oracle Essbase Technical Reference*
ESSCMD	LOADDB or SELECT	*Oracle Essbase Technical Reference*

➤ To configure a database to start automatically when its parent application starts, use a tool:

Tool	Topic	Location
Administration Services	Configuring Databases to Start Automatically	*Oracle Essbase Administration Services Online Help*
MaxL	**alter database enable autostartup**	*Oracle Essbase Technical Reference*

Stopping a Database

Stopping a database unloads all data from memory and commits any updated data to disk. If a database is stopped and a user attempts to retrieve data from it, the database starts on Essbase Server automatically, without any explicit commands issued.

When you stop a database, transactions may be currently running. If you stop a database using any of the proper methods (see the following table), the database does not stop if a calculation or data load is in progress. Instead, Essbase displays a message in the server console window.

If you stop the Agent by closing the server console window or by pressing Ctrl+C, the database stops, and the next time you start the database, Essbase rolls back any transactions that were in progress. See "Rollback with Committed Access" on page 783 or "Rollback with Uncommitted Access" on page 785.

➤ To stop a database, use a tool:

Tool	Topic	Location
Agent	STOP *appname*	Enter the Agent command at the command prompt in the Essbase Server console window.
Administration Services	Stopping Databases	*Oracle Essbase Administration Services Online Help*
MaxL	**alter application unload database**	*Oracle Essbase Technical Reference*
ESSCMD	UNLOADDB	*Oracle Essbase Technical Reference*

Managing Ports

The Agent enables you to manage ports on Essbase Server.

Viewing a List of Users and Available Ports

You can view a list of users that are connected to Essbase Server at any given time, the number of ports available, and the number of connections.

➤ To view a list of users connected to Essbase Server, use a tool:

Tool	**Topic**	**Location**
Agent	USERS	Enter the Agent command at the command prompt in the Essbase Server console window.
Administration Services	Viewing Essbase Server Users and Groups	*Oracle Essbase Administration Services Online Help*
MaxL	**display user**	*Oracle Essbase Technical Reference*
ESSCMD	LISTUSERS	*Oracle Essbase Technical Reference*

➤ To view the number of ports installed on Essbase Server, as well as the number of ports in use, use a tool:

Tool	**Topic**	**Location**
Agent	PORTS	Enter the Agent command at the command prompt in the Essbase Server console window.
Administration Services	Checking Available Ports Also see About Essbase Connections and Ports	*Oracle Essbase Administration Services Online Help*
MaxL	**display user**	*Oracle Essbase Technical Reference*
ESSCMD	LISTUSERS	*Oracle Essbase Technical Reference*

Specifying Nondefault Port Values

To change the default port values used by the Agent, you must set one or more of these configuration settings:

- AGENTPORT specifies the port that the Agent uses.
- SERVERPORTBEGIN specifies the first port number the Agent on a computer tries to use for its first server process.

- SERVERPORTEND specifies the highest value the Agent tries to use for a port when it tries to start a server process. If the value is unavailable, the server process fails.
- PORTINC specifies the value of the increment in between port numbers used by the Agent.

You may want to change the default value. Two possible reasons:

- The default value specifies a port number already in use.
- You may want to install a second instance of Essbase Server on a computer to facilitate testing. You can use the Oracle's Hyperion Enterprise Performance Management System Configurator to assign the second Essbase Server instance to a different port than the first. See the *Oracle Hyperion Enterprise Performance Management System Installation and Configuration Guide*.

Changing Port Default Values

If you need to change one or more of the default values associated with Agent and server ports, see the *Oracle Essbase Technical Reference* for the correct configuration setting to change.

Viewing Port Statistics

You can enable Essbase to log, at a specified interval, the number of ports being used. By analyzing the information in the log, you can monitor port utilization and identify a need for more ports before users are unable to connect.

To enable Essbase Server to check port use statistics and write those statistics to the Essbase Server log, use the PORTUSAGELOGINTERVAL configuration setting:

```
PORTUSAGELOGINTERVAL n
```

where *n* represents the number of minutes between each check of the number of ports in use. The value of *n* can be any whole number from 1 to 60, with 5 being the recommended minimum and default value. Essbase ignores any portion of a nonwhole number. For example, Essbase evaluates 2.5 as 2 minutes. Statistics are written to the Essbase Server log immediately after each check. The log file will resemble the following output:

```
[Mon Apr 22 00:48:50 2003]Local/ESSBASE0///Info(1056214)
[3] ports in use, [10] ports allowed
```

See the *Oracle Essbase Technical Reference*.

Managing Essbase Administration Server Communication Ports

Essbase Administration Server has several configurable communication ports, which are different from Essbase Server ports. See "About Essbase Administration Server" on page 112.

Controlling Query Size and Duration

Users may unintentionally request information that is so large or so complex to retrieve that the query will slow performance or fail to complete properly. Use the QRYGOVEXECTIME and QRYGOVEXECBLK configuration settings, referred to as query governors, to control query size or duration:

- `QRYGOVEXECTIME [`*`appname`*` [`*`dbname`*`]] `*`n`*

  Limits the time Essbase Server allows a query to run before terminating the query.

- `QRYGOVEXECBLK [ `*`appname`*` [`*`dbname`*`]] `*`n`*

  Limits the number of blocks a query can access before terminating the query.

You can apply these settings to all the applications and databases on Essbase Server, to all the databases on a single application, or to one database. See the *Oracle Essbase Technical Reference*.

Increasing Agent Connections to Essbase Server

Increasing the maximum possible threads between Essbase Server and the Agent allows multiple users to log on and connect to an application and database at the same time.

Use the AGENTTHREADS and AGTSVRCONNECTIONS configuration settings to control the maximum number of threads created to perform the initial connection to Essbase Server:

- `AGENTTHREADS `*`maximum_number_of_threads`*
- `AGTSVRCONNECTIONS `*`maximum_number_of_threads`*

  Keep the *`maximum_number_of_threads`* value for AGTSVRCONNECTIONS equal to or less than the value for AGENTTHREADS to avoid wasting resources. Each connection requires one thread each from the server and Agent, so there is no need for higher values for AGTSVRCONNECTIONS. The default value for each setting is 5, the minimum is 1, and the maximum is 500. See the *Oracle Essbase Technical Reference*.

Note:

All requests for information after initial connection and before disconnection are handled by a different set of server threads, whose maximum number is controlled by the SERVERTHREADS configuration parameter.

Limiting the Number of User Sessions

Use the MAXLOGIN configuration setting to limit the maximum number of simultaneous user session connections to Essbase Server. This number includes multiple instances of the same user. For example, one user with five open Excel worksheets connected to the same Essbase Server uses one port but five sessions.

You can adjust the value of MAXLOGIN to match computer resources or to more closely manage concurrent ports and user sessions. A concurrent port is used for each unique combination of client computer, Essbase Server, and login name. See the *Oracle Essbase Technical Reference*.

Note:

User sessions use the threads whose maximum is controlled by the SERVERTHREADS configuration setting and are not related to the threads whose maximum is controlled by AGENTTHREADS and AGTSVRCONNECTIONS.

44 Managing Applications and Databases

In This Chapter

Also see the *Oracle Hyperion Enterprise Performance Management System Backup and Recovery Guide*.

Understanding Applications and Databases

An application is a management structure that contains one or more Essbase databases and related files. Essbase applications and databases usually reside on the Essbase Server. The server computer can store multiple applications.

An Essbase database is a data repository that contains a multidimensional data storage array. A multidimensional database supports multiple views of data so that users can analyze the data and make meaningful business decisions.

Files that are related to Essbase databases are called *artifacts* or *objects*. Database artifacts perform actions against one or more Essbase databases, such as defining calculations or reporting against data. By default, artifacts are stored in their associated database folder on the server. Some artifacts also can be saved to a client computer or to other available network directories. See "Understanding How Essbase Files Are Stored" on page 710.

The common types of database artifacts in Essbase:

- A database outline (a storage structure definition)
- Data sources
- Rules for loading data and building dimensions dynamically (rules files)
- Scripts that define how to calculate data (calculation scripts)
- Scripts that generate reports on data (report scripts)
- Security definitions
- Security filters

- LROs
- Partition definitions

Some of these artifacts are optional, such as calculation scripts, filters, and LROs.

For a complete description of each database artifact, see "Understanding Database Artifacts" on page 116.

Understanding How Essbase Files Are Stored

Essbase installation files and files that are created when using Essbase are stored in the following locations:

- `ESSBASEPATH`: The Essbase installation directory and all subdirectories under it (with the exception of the `app` directory).

  In a default installation, the `ESSBASEPATH` directories are:

  - For Essbase Server: `Hyperion/products/Essbase/EssbaseServer`
  - For Essbase Client: `Hyperion/products/Essbase/EssbaseClient`

  For a list of directories that are created under `ESSBASEPATH`, see the *Oracle Hyperion Enterprise Performance Management System Installation and Configuration Guide*.
- `ARBORPATH`: The `app` directory where Essbase application files (as they are created) and sample applications and databases (provided with Essbase) are stored.

  In a default installation, `ARBORPATH` is:

  `Hyperion/products/Essbase/EssbaseServer/app`
- `HYPERION_HOME`: The directory under which all Oracle Hyperion Enterprise Performance Management System products are installed.

  In a default installation, `HYPERION_HOME` is:

  `Hyperion`
- `HYPERION_HOME/common`: The directory where common internal and third-party components are stored.
- `HYPERION_HOME/logs`: The directory where log files are stored.

Server Software File Types

This table lists the types of Essbase files stored in the *`ESSBASEPATH`*`/bin` directory:

Table 71 Essbase File Types in *`ESSBASEPATH`*`/bin`

File Extension	Description
bak	Backup of security file
bnd	Microsoft ODBC file for SQL Interface installation using a DB2 database

File Extension	Description
cfg	Essbase Server configuration file
cnt	Online help contents file
cpl	Microsoft ODBC driver for Windows platforms
dll	Microsoft Windows Dynamic Link Library
eqd	Query Designer files
exe	Executable file
hlp	Online help file
lck	Lock file
lic	License information file for ODBC
pl	Sample Perl script
pm	Perl Module
mdb	Message database file
sec	Security file
sl	HP-UX shared library file
so	Solaris shared library file
xll	Spreadsheet Add-in

Application and Database File Types

The following table lists the file types that Essbase uses to store applications, databases, and their related artifacts.

Table 72 Essbase File Types for Applications and Databases

File Extension	Description
alg	Spreadsheet audit historical information
apb	Backup of application file
app	Application file, defining the name and location of the application and other application settings
arc	Archive file
atx	Spreadsheet audit transaction
chg	Outline synchronization change file
csc	Essbase calculation script

File Extension	Description
db	Database file, defining the name, location, and other database settings
dbb	Backup of database file
ddb	Partitioning definition file
ddm	Temporary partitioning file
ddn	Temporary partitioning file
esm	Essbase kernel file that manages pointers to data blocks and contains control information used for database recovery
esr	Temporary database root file
esn	Temporary Essbase kernel file
ind	Essbase index file
inn	Temporary Essbase index file
log	Server or application log
lro	LRO file linked to a data cell
lst	Cascade table of contents or list of files to back up
mxl	MaxL script file (saved in Administration Services)
ocl	Database change log
ocn	Incremental restructuring file
oco	Incremental restructuring file
olb	Backup of outline change log
olg	Outline change log
otl	Essbase outline file
otm	Temporary Essbase outline file
otn	Temporary Essbase outline file
oto	Temporary Essbase outline file
pag	Essbase database data (page) file
pan	Temporary Essbase database data (page) file
rep	Essbase report script
rul	Essbase rules file
scr	Essbase ESSCMD script

File Extension	Description
sel	Saved member select file
tct	Essbase database transaction control file that manages all commits of data and follows and maintains all transactions
tcu	Temporary database transaction control file
trg	Trigger definition file. XML (Extensible Markup Language) format
txt	Text file, such as a data file to load or a text document to link as a LRO
xcp	Exception error log
xls	Microsoft Excel file

API File Types

The following table lists the types of Essbase files stored in the *ESSBASEPATH*/`api` subdirectories:

Table 73 Essbase File Types in the **`api`** Directory

File Extension	Description
a	UNIX static library file
bas	Microsoft Visual Basic program source file, containing header definitions for the Essbase API
h	C or C++ header file, containing header definitions for the Essbase API
lib	C or C++ program library
np	Named Pipes network library
tcp	TCP/IP network library

Managing Applications, Databases, and Database Artifacts

This section explains how to manage applications, databases, and database artifacts.

For a description of Essbase applications, databases, and database artifacts, see "Understanding Applications and Databases" on page 116 and "Understanding Database Artifacts" on page 116.

Strategies for Backing Up and Recovering Databases

Regular Essbase backups, which should be integrated into production server maintenance, are key to database maintenance. Backup frequency should be determined by the volatility of the database and server environment and the need for rapid database restoration (should a server interruption occur).

To back up and restore block storage databases, you can use either of the following methods:

- Automated database backup and restore and transaction logging and replay

  Backup and restore provides the equivalent functionality of manually backing up and restoring a database. When a backed-up database is restored, transactions that occurred after the backup procedure are not recovered. However, with transaction logging and replay, post-backup transactions are captured and can be replayed. Thus, a backed-up database can be recovered to the most-recent state before the interruption occurred.

  The use of the database backup and restore and transaction logging and replay features eliminates the need for various manual steps and, therefore, enables administrators to back up and recover databases more efficiently. Oracle recommends incorporating these features in your backup and recovery strategy.

- Manual backup and restore

  Essbase customers who have designed a backup and restore strategy that uses manual procedures and who do not need the functionality of transaction logging and replay can continue using their manual strategy.

To back up and restore aggregate storage applications, you must use manual procedures.

See the *Oracle Hyperion Enterprise Performance Management System Backup and Recovery Guide*.

Using the File System to Manage Applications and Databases During Backup

You should not use the platform file system to copy, move, rename, or delete applications and databases. When an application or database is altered through the file system, the Essbase security file is unable to recognize the changes. This situation creates a mismatch between what actually exists on the hard drive and what exists according to Essbase.

Caution!

Do not move, copy, modify, or delete any of these files: `ess`*`n`*`.ind`, `ess`*`n`*`.pag`, *`dbname`*`.ind`, *`dbname`*`.esm`, or *`dbname`*`.tct`. Doing so may result in data corruption.

Certain application and database files can be managed successfully through the file system:

- Rules files for dimension builds and data loads (`.rul`)
- Data load or dimension build files
- Calculation scripts (`.csc`)
- Report scripts (`.rep`)
- MaxL scripts (`.mxl` or any extension)

➤ To copy or move an outline file (`.otl`), you must use Administration Services. See "Copying Outlines" in the *Oracle Essbase Administration Services Online Help*.

The only time the file system should be used to manage applications and databases is during backup, when the entire directory for an application or database is copied and stored elsewhere. See the *Oracle Hyperion Enterprise Performance Management System Backup and Recovery Guide*.

Monitoring Applications

Each application that is loaded is an open task or process in the operating system. On Windows platforms, the application is displayed in a command-line window. On UNIX platforms, the application server is a child process of `ESSBASE`. When the application starts, ESSBASE starts the `esssvr` process. See "Starting an Application" on page 701.

On Windows platforms, when an application starts, a new icon is displayed in the taskbar. Double-click the icon to view the server window.

Essbase Server records application-level activities in an application log. See "Using Essbase Logs" on page 729.

➤ To view application activities as they occur, use a tool:

Tool	Instruction
On Windows platforms, use the application-process window	Select the command-line window that bears the name of the application.
UNIX	`tail -f` *`logfile`*

Using Essbase to Manage Applications and Databases

This section describes managing applications and databases.

Viewing Applications and Databases

When you start Administration Services Console, the Enterprise View tree is displayed in the navigation panel. Enterprise View is a graphical tree view of the Essbase environment. It displays the Administration Servers and Essbase Servers that you select. Your view of the Essbase environment may look different from that of other administrators.

Applications and databases, and their associated artifacts, are represented as nodes beneath the Essbase Server node. Artifacts are grouped into container nodes. For example, individual applications are contained in the Applications node, and databases are contained in the Databases container node. If sample applications and databases are installed with Essbase Server, they appear in Enterprise View along with your organization's applications and databases.

See "About Enterprise View" in the *Oracle Essbase Administration Services Online Help*.

➤ To create an application, see “Creating Applications” in the *Oracle Essbase Administration Services Online Help*.

Copying or Migrating Applications

You can copy an application to any Essbase Server to which you have appropriate access. You can copy (migrate) an entire application to another Essbase Server, or you can copy an application on the same Essbase Server. For example, you may need to migrate an entire application from a development server to a production server. Or, you may want to copy an application on the same server for testing or for backup purposes.

Essbase copies applications differently, depending on whether you are copying to the same Essbase Server or to a different Essbase Server. When you migrate applications, you can select the artifacts to migrate, such as calculation scripts, report scripts, rules files, custom-defined macros and functions, substitution variables, and filters. You can also specify how user and group security is migrated.

Administration Services provides a Migration Wizard that helps you migrate applications. See “Migration Wizard” in *Oracle Essbase Administration Services Online Help*.

➤ To copy an application, use a tool:

Tool	Topic	Location
Administration Services	Copying Applications	*Oracle Essbase Administration Services Online Help*
MaxL	**create application as**	*Oracle Essbase Technical Reference*
ESSCMD	COPYAPP	*Oracle Essbase Technical Reference*

Renaming Applications

When you rename an application, the application and its associated directory (`*ARBORPATH*/app/*appname*`) are renamed. All artifacts within the application (for example, databases or calculation scripts) with the same name as the application are not renamed. Before you rename an application, see “Naming Restrictions for Applications and Databases” on page 1059.

➤ To rename an application, use a tool:

Tool	Topic	Location
Administration Services	Renaming Applications	*Oracle Essbase Administration Services Online Help*
MaxL	**alter application**	*Oracle Essbase Technical Reference*
ESSCMD	RENAMEAPP	*Oracle Essbase Technical Reference*

Deleting Applications

When you delete an application, all artifacts within the application also are deleted. The `ARBORPATH/app/appname` directory and all files in the directory are deleted.

➤ To delete an application, use a tool:

Tool	Topic	Location
Administration Services	Deleting Applications	*Oracle Essbase Administration Services Online Help*
MaxL	**drop application**	*Oracle Essbase Technical Reference*
ESSCMD	DELETEAPP	*Oracle Essbase Technical Reference*

Copying Databases

You can copy a database in an application to any Essbase Server and application to which you have appropriate access. You can copy (migrate) an entire database to another Essbase Server, or you can copy a database on the same Essbase Server. For example, you may need to migrate an entire database from a development server to a production server. Or, you may want to copy a database on the same server for testing or for backup purposes. Essbase copies databases differently depending on whether you are copying to the same Essbase Server or to a different Essbase Server. See “Copying Databases” in *Oracle Essbase Administration Services Online Help*.

Administration Services provides a Migration Wizard that helps you migrate applications and databases. See “Migration Wizard” in *Oracle Essbase Administration Services Online Help*.

When you copy a database, all files associated with the database, except data files (`.pag` and `.ind`), are copied to the destination application. Before copying, make sure you have enough disk space to contain a full copy of the database and its related files.

➤ To copy a database, use a tool:

Tool	Topic	Location
Administration Services	Copying Databases	*Oracle Essbase Administration Services Online Help*
MaxL	**create database as**	*Oracle Essbase Technical Reference*
ESSCMD	COPYDB	*Oracle Essbase Technical Reference*

Note:

Essbase allows copying a non-Unicode database to a Unicode application; however, copying a Unicode database to a non-Unicode application is not allowed.

Renaming Databases

When you rename a database, the database and its associated directory (`ARBORPATH/app/appname/dbname`), and the outline file (`.otl`) are renamed. All other artifacts in the database (for example, calculation scripts) with the same name as the database are not renamed.

➤ To rename a database, use a tool:

Tool	Topic	Location
Administration Services	Renaming Databases	*Oracle Essbase Administration Services Online Help*
MaxL	**alter database**	*Oracle Essbase Technical Reference*
ESSCMD	RENAMEDB	*Oracle Essbase Technical Reference*

Deleting Databases

When you delete a database, all artifacts within it are also deleted. The `ARBORPATH/app/appname/dbname` directory and all files located in the directory are deleted.

➤ To delete a database, use a tool:

Tool	Topic	Location
Administration Services	Deleting Databases	*Oracle Essbase Administration Services Online Help*
MaxL	**drop database**	*Oracle Essbase Technical Reference*
ESSCMD	DELETEDB	*Oracle Essbase Technical Reference*

Using Essbase to Manage Artifacts

This section describes copying, renaming, and deleting artifacts, such as outlines, calculation scripts, report scripts, rules files, and data sources. See "Understanding Database Artifacts" on page 116.

Caution!

The only time the file system should be used to manage applications is during backup, when the entire directory for an application or database is copied and stored elsewhere.

Copying Artifacts

You can copy any database artifact, except an outline, to another application, database, server, or client location. For instructions on copying outlines, see "Creating and Editing Outlines" on page 128.

➤ To copy an artifact, use a tool:

Tool	Topic	Location
Administration Services	Topics on copying the specific artifact; for example, Copying a Rules File	*Oracle Essbase Administration Services Online Help*
MaxL	**alter object**	*Oracle Essbase Technical Reference*
ESSCMD	COPYOBJECT	*Oracle Essbase Technical Reference*

Renaming Artifacts

You can rename any artifact except an outline. An outline always has the same name as the database, so you must rename the database to rename the outline.

➤ To rename an artifact, use a tool:

Tool	Topic	Location
Administration Services	Topics on renaming the specific artifact; for example, Renaming a Rules File	*Oracle Essbase Administration Services Online Help*
MaxL	**alter object**	*Oracle Essbase Technical Reference*
ESSCMD	RENAMEOBJECT	*Oracle Essbase Technical Reference*

Deleting Artifacts

You can delete any artifact except an outline. An outline is a required part of a database, so you must delete the database to delete the outline.

➤ To delete an artifact, use a tool:

Tool	Topic	Location
Administration Services	Topics on deleting the specific artifact; for example, Deleting a Rules File	*Oracle Essbase Administration Services Online Help*
MaxL	**drop object**	*Oracle Essbase Technical Reference*
ESSCMD	DELETE command for the artifact to delete	*Oracle Essbase Technical Reference*

Locking and Unlocking Artifacts

Essbase uses a checkout facility for database artifacts to ensure that only one user modifies an artifact at one time. This section describes how to lock and unlock artifacts, with the exception of outlines. See "Locking and Unlocking Outlines" on page 129.

Note:

Locking artifacts is not the same as locking data blocks. The Essbase kernel handles locking for data blocks but not for artifacts. See "Data Locks" on page 780.

By default, whenever you open a database artifact, Essbase prompts you to lock the artifact. You can change this default behavior for some artifacts; see "Setting Essbase Default Options" in *Oracle Essbase Administration Services Online Help*.

When an artifact is locked, Essbase does not allow other users to save over, rename, or delete the artifact. You can open a locked artifact and edit it, but you cannot save over the existing artifact. To save changes made to a locked artifact, save the modified artifact to a different location. You can execute and copy locked artifacts.

Unlocking Artifacts

You can unlock artifacts according to your permissions. In Administration Services, you can view all artifact locks for an Essbase Server, application, or database.

➤ To unlock an artifact, use a tool:

Tool	Topic	Location
Administration Services	Locking and Unlocking Objects	*Oracle Essbase Administration Services Online Help*
MaxL	**alter object**	*Oracle Essbase Technical Reference*
ESSCMD	UNLOCKOBJECT	*Oracle Essbase Technical Reference*

Migrating Applications Using Administration Services

Using Administration Services, you can migrate applications to any Essbase Server to which you have appropriate access, regardless of platform. For example, you may need to migrate an application from a development server to a production server. When you migrate applications, you can select the artifacts to migrate, such as calculation scripts, report scripts, rules files, custom-defined macros and functions, substitution variables, and filters. You can also specify how user and group security is migrated.

➤ To migrate an application, see "Copying Applications" in *Oracle Essbase Administration Services Online Help*.

Porting Applications Across Platforms

Essbase runs on multiple platforms, including Windows and UNIX. For a list of supported platforms, see the *Oracle Hyperion Enterprise Performance Management System Installation Start*

Here. For information on how to install and configure Essbase on each platform, see the *Oracle Hyperion Enterprise Performance Management System Installation and Configuration Guide*.

After you create an application, you may want to port the application to a server that runs a different operating system. This section describes how to port an application to another Essbase computer.

Porting Essbase applications across servers involves these steps:

1. Identifying compatible files
2. Checking file names
3. Transferring compatible files
4. Reloading the database

Identifying Compatible Files

If you are porting an Essbase application to a server that uses a different operating system, you must identify which Essbase files are compatible with the new operating system.

The following file types are compatible between operating systems:

- Text files. The Essbase text files are calculation scripts (`.csc`) and report scripts (`.rep`), and any MaxL or ESSCMD scripts you have developed. Also, data files can be text files.
- Rules files. These files are binary files, but they are compatible between operating systems. Rules files have the extension `.rul`.
- Outline files. These files are binary files, but they are compatible between operating systems. Outline files have the extension `.otl`.

The following file types are incompatible between operating systems and must be redefined or reloaded on the new server:

- Database files with the extensions `.db` and `.dbb`
- Data files with the extension `.pag`
- Index files with the extension `.ind`
- Security files with the extension `.sec`
- Application files with the extensions `.app` and `.apb`
- Essbase Kernel files with the extension `.esm`

Note:

If you are using the Linked Reporting Objects feature, you must relink any files or cell notes on the new server. For a comprehensive discussion of how LROs are used, see Chapter 11, "Linking Objects to Essbase Data."

Checking File Names

When transferring files to a UNIX system, be aware of the case-sensitivity of file names. UNIX is a case-sensitive operating system, and files are recognized only if file names are in the correct case. For example, in certain MaxL and ESSCMD operations, you must specify a file name, and the file name must be entered in the correct case.

The Essbase system files use the following naming conventions on UNIX systems:

- Executable files have no extension and are uppercase (for example, `ESSBASE`, `ESSCMD`).
- Static library files have the file extension `.a` and are in lowercase (for example, `libessnet.a`).
- Shared library files have the file extension `.sl` on HP-UX, `.so` on Solaris, and `.a` on AIX. These file names are in lowercase (for example, `libesscur.sl`).
- Security files have the file extension `.sec` and are in lowercase (for example, `essbase.sec`).
- Message database files have the file extension `.mdb` and are in lowercase (for example, `essbase.mdb`).
- Online help files have the file extension `.hlp` and are in lowercase (for example, `esscmd.hlp`).

Essbase files on UNIX systems are capitalized with *proper* case—the first letter is uppercase, and the remaining letters are lowercase. Table 74 gives examples of names for different file types:

Table 74 File Naming Examples for UNIX

File Type	Example
Database files	`Mydb.db`
Data files	`Mydb.pag`
Index files	`Mydb.ind`
Outline files	`Mydb.otl`
Rules files	`Atlanta.rul`
Data files to load	`Atlanta.txt`
Calculation scripts	`Mycalc.csc`
Report scripts	`Myrepo.rep`
Archive files	`Mydb.arc`
Application logs	`Myapp.log`

Note:

The application name is an exception to the above rule. The application name can be in lowercase.

Table 75 lists several examples of valid and invalid file names on UNIX systems:

Table 75 Valid and Invalid File Names on UNIX

Valid File Names	Invalid File Names
`Model.csc`	`MODEL.CSC`
`Monthly.rep`	`Monthly.Rep`
`Forecast.otl`	`forecast.otl`
`Actuals.rul`	`AcTuAlS.rUl`
`My_File.txt`	`My_File.Txt`

Note:

Essbase does not allow long file names for applications, databases, calculation scripts, reports, and other database files. All file names for artifacts you create must conform to the Windows 8.3 convention.

Transferring Compatible Files

If two servers are connected, you can create the application and database directories on the new server and use either FTP (File Transfer Protocol) or Administration Services to transfer the compatible application files. If the servers are not connected, you must redefine server information on the new server before reloading the database.

Using FTP to Transfer Files

Using FTP, you can transfer files directly between operating systems. You should transfer only the files that are compatible between operating systems, and you should transfer the files in binary mode.

If you have files with the wrong case on a UNIX server, Administration Services can see them but cannot open them. After you use FTP to transfer files, rename the files on the server to ensure that they are capitalized in the proper case. Alternatively, you can use FTP to rename the file when you transfer the file. For example:

```
ftp>put oldfile Newfile
```

Using Administration Services to Transfer Files

Using Administration Services, you can transfer files from the client computer to the server in the following ways:

- As part of an application migration. See “Migration Wizard” in *Oracle Essbase Administration Services Online Help*.

- As part of a database migration. See “Copying Databases” in *Oracle Essbase Administration Services Online Help*.
- One artifact at a time. See the topic for the individual artifact; for example, “Copying Rules Files,” in *Oracle Essbase Administration Services Online Help*.

Redefining Server Information

If the server you are porting to is not connected to the existing server, you must redefine some information on the new server.

➤ To redefine server information:

1 **To create users and specify their permissions, use Administration Services on the new server.**

   See “Granting Permissions to Users and Groups in Native Security Mode” on page 628.

2 **To create the applications and databases that you want to port, use Administration Services on the new server.**

   See Chapter 6, “Creating Applications and Databases.”

3 **Copy the outline files (`.otl`) for the databases that you want to port from the old server to the same directory location on the new server. Ensure that the application is not running while you copy these files.**

   See “Creating and Editing Outlines” on page 128.

4 **Copy compatible files from the old server to the new server.**

   See “Identifying Compatible Files” on page 721.

5 **Reload the database.**

   See “Reloading the Database” on page 724.

Reloading the Database

Database files, such as `.db`, `.pag`, `.esm`, and `.ind`, are not compatible between operating systems. If you port an application to a server on a different operating system, you must repopulate the database by reloading the data from a data file and a rules file (if applicable). One way you can reload is to export the data to a text file, transfer the text file to the new server, and then use the text file to load data. After the load is complete, calculate the new database.

45 Monitoring Data, Applications, and Databases

In This Chapter

Monitoring Data Changes Using Triggers

The triggers feature provided by Essbase enables efficient monitoring of data changes in a database.

If data breaks rules specified in a trigger, Essbase can log relevant information in a file or, for some triggers, can send an e-mail alert (to a user or system administrator); for example, to notify the sales manager if, in the Western region, sales for a month fall below sales for the equivalent month in the previous year.

There are two types of triggers. On-update triggers are activated during the update process, when the block containing the data referenced by the trigger is brought into memory. After-update triggers are activated after the update transaction is complete.

Note:

On-update triggers are supported only on block storage databases.

Administering Triggers

To administer triggers, a user must have Database Manager security privilege. Essbase monitors and potentially activates triggers during the following activities:

- Data load
- Calculation
- Lock and send from Oracle Essbase Spreadsheet Add-in (available only for on-update triggers on block storage databases)

Essbase does not activate triggers during a database restructure.

➤ To create, change, display, and delete triggers, use a tool:

Tool	Topic	Location
Administration Services	Creating Triggers Editing Triggers Viewing Triggers Enabling and Disabling Triggers Viewing Trigger Spool Files Deleting Triggers	*Oracle Essbase Administration Services Online Help*
MaxL	**create trigger** **create or replace trigger** **alter trigger** **display trigger** **drop trigger**	*Oracle Essbase Technical Reference*

You can see information on enabled and disabled triggers in the application log file when you start Essbase Server.

Creating On-Update Triggers

When creating on-update triggers, consider the following information:

- You must use a symmetric WHERE clause when defining the area specification for a trigger. See the MDX documentation in the MaxL section of the *Oracle Essbase Technical Reference*.
- To enable Essbase to send e-mail alerts, you must have JVM installed on your system.

  Note:

  E-mails related to Unicode-mode applications are encoded in UTF-8 and require a UTF-8-capable e-mail reader.
- You cannot define a trigger that requires data from the following Essbase features:
  - Dynamic Calc
  - Hybrid Analysis
  - Partitioning
- You can specify whether all trigger data values are stored in the spool file or whether only the most current values are stored (for example, use the a log_value parameter on the MaxL **create trigger** statement or **create and replace trigger** statement). If the log_value parameter is set to ON, both the new value and old value are logged to the spool file. If the log_value parameter is set to OFF, values are not logged to the spool file. The log_value parameter is active only for data load and lock-and-send activities.

Consider data security when sending e-mail alerts. When Essbase activates a trigger and sends an e-mail, it cannot check whether the e-mail recipient is authorized to see the data referenced by the trigger condition.

Avoid referencing a sparse dimension member in a trigger condition. When Essbase executes a trigger on one data block, it has to read another data block that may or may not be updated. Depending on the state of the second block, Essbase may activate the trigger in error.

The following example is based on the Sample.Basic database. Assume that East and West are sparse dimensions and that the following statement defines the trigger condition:

```
FIX (East, West, Sales, Jan, Actual, Cola)
IF ((Sales -> East + Sales -> West) > 20)
EMAIL sales@.com
```

Assume that the following Sales data values are stored for East and West:

- Sales -> East = 5
- Sales -> West = 6

Now assume that a user does a lock-and-send request to update the data values:

- Sales -> East = 15
- Sales -> West = 3

When Sales->East is updated to 15, temporarily, the database contains the following values:

- Sales -> East = 15
- Sales -> West = 6

So Essbase activates the trigger because 15 plus 6 is greater than 20. Subsequently, Essbase updates Sales -> West to 3. Now the data does not meet the trigger condition, because 15 + 3 < 20. However, Essbase has already activated the trigger.

When a data load is followed by a calculation, if both the loaded data and the calculated data meet the trigger condition, Essbase activates the trigger twice, once on the data load and once on the calculation.

Creating After-Update Triggers

When creating after-update triggers, consider the following information:

- You must use a symmetric WHERE clause when defining the area specification for a trigger. See the MDX documentation in the MaxL section of the *Oracle Essbase Technical Reference*.
- You cannot define a trigger that requires data from the Essbase Hybrid Analysis feature.
- Spreadsheet lock-and-send operations do not activate after-update triggers.

Effect of Triggers on Performance and Memory Usage

Depending on the number of enabled triggers in a database, there may be a small decrease in performance of calculation and data load. You can control the maximum amount of memory used by the triggers feature by specifying the TRIGMAXMEMSIZE configuration setting in the `essbase.cfg` file. By default, TRIGMAXMEMSIZE is set to 4096 bytes. Choosing to log information in a file, rather than sending e-mail, may improve calculation and data load performance by reducing network traffic.

Trigger Examples

The following examples are based on the Sample.Basic database.

Note:

You cannot define a trigger that requires data from hybrid analysis members. You cannot define an on-update trigger that requires data from Dynamic Calc members or from members from another partition.

Example 1, Tracking Sales for January

Example 1 tracks the Actual, Sales value for the following month, product, and region:

- January (Year dimension member Jan)
- Colas (Product dimension member 100)
- Eastern region (Market dimension member East)

When the member being calculated is Jan, and when the Actual, Sales value of Colas for January exceeds 20, the example sends an e-mail to two e-mail accounts.

```
create trigger Sample.Basic.Trigger_Jan_20
where "(Jan,Sales,[100],East,Actual)"
when Jan > 20 and is(Year.currentmember,Jan) then
mail ([Docs.Company.com],[trgsales@company.com],
  [inventory@company.com],
[Mail sent by trigger_Jan_20])
end;
```

Example 2, Tracking Sales for Quarter 1

Example 2 tracks the Actual, Sales value for the following months, product, and region:

- January, February, and March (the children of Year dimension member Qtr1)
- Colas (Product dimension member 100)
- Eastern region (Market dimension member East)

When the member being calculated is Jan, Feb, or Mar, and when the Actual, Sales value of Colas for the month January, February, or March exceeds 20, the example logs an entry in the file Trigger_Jan_Sales_20, Trigger_Feb_Sales_20, or Trigger_Mar_Sales_20. On subsequent trigger activations, both old and new log values are retained in the log files.

```
create or replace trigger Sample.Basic.Trigger_Qtr1_Sales
log_value on
Where "(crossjoin(Qtr1.children, {(Measures.Sales, [100],
  East, Scenario.Actual)}))"
When Year.Jan > 20 and is(Year.currentmember, Jan) then
  spool Trigger_Jan_Sales_20
When Year.Feb > 20 and is(Year.currentmember, Feb) then
  spool Trigger_Feb_Sales_20
When Year.Mar > 20 and is(Year.currentmember, Mar) then
  spool Trigger_Mar_Sales_20
end;
```

Example 3, Tracking Inventory Level

Example 3 tracks the inventory level for the following product, region, and months:

- Colas (product 100)
- Eastern region (market East)
- January, February, and March (the children of Qtr1)

The trigger is activated after the update action is complete. If the inventory of Colas in the eastern region falls below 500,000, the example logs an entry in the file Inventory_East.

```
create after update trigger Sample.Basic.Inventory_east
where "(crossjoin ({children([Qtr1])},
{([Market].[East], [Product].[100],
  [Inventory].[Ending Inventory])}))"
when [Ending Inventory] < 500000 then
  spool Inventory_East
end;
```

Using Essbase Logs

This topic describes the logs that Essbase Server creates to record information about server, application, and database activities. Table 76 briefly describes each log.

Note:

Log files (except for Essbase Server log files) may be located under *ARBORPATH*/app/*appname*, depending on the value of the DEFAULTLOGLOCATION configuration parameter.

Table 76 Summary of Logs

Type of Log	Location of Log	Information Included
Essbase Server log	*HYPERION_HOME*/logs/essbase/ESSBASE.LOG **Note:** If your installation contains multiple instances of Essbase Server, each instance has its own log file directory, as controlled by the InstanceID configuration setting. For example: if InstanceID is set to inst1, the logging directory for that instance is	Server activities and errors

Type of Log	Location of Log	Information Included
	HYPERION_HOME/logs/essbase_inst1.	
Application log	*HYPERION_HOME*/logs/essbase/app/*appname*/ *appname*.LOG	Application activities and errors
Query log	*HYPERION_HOME*/logs/essbase/app/*appname*/*dbname*/*dbname*00001.qlg	Query patterns of Essbase database retrievals
Outline change log	*HYPERION_HOME*/logs/essbase/app/*appname*/*dbname*/*dbname*.olg	Changes to the outline
Exception log	One of these locations: *ARBORPATH*/log00001.xcp *HYPERION_HOME*/logs/essbase/app/log00001.xcp *HYPERION_HOME*/logs/essbase/app/*appname*/log00001.xcp *HYPERION_HOME*/logs/essbase/app/*appname*/*dbname*/log00001.xcp	Errors that result when Essbase Server stops abnormally
Dimension build and data load error logs	One of these locations (see Table 85 on page 754): *ARBORPATH*/client/dataload.err *HYPERION_HOME*/logs/essbase/app/*appname*/*appname*.log	Errors from a dimension build or a data load

This topic describes the information written to a log and explains how you can use that information to maintain, tune, or troubleshoot Essbase Server.

For information about Administration Services logs, see "About the Administration Server Log" in *Oracle Essbase Administration Services Online Help*.

Essbase Server and Application Logs

Essbase Server writes, or logs, activities that occur in Essbase Server and applications in text files with a `.log` extension.

The Essbase Server log, named `ESSBASE.LOG`, is located in this directory in a standard Essbase installation:

HYPERION_HOME/logs/essbase/ESSBASE.LOG

Each application on Essbase Server has its own application log, named after the application. For example, the log file for the Sample application is named `sample.log`. In a standard Essbase installation, application logs are located in one of these directories, depending on the value of the DEFAULTLOGLOCATION configuration parameter:

- *HYPERION_HOME*/logs/essbase/app/*appname*
- *ARBORPATH*/app/*appname*

Information in application logs can help you pinpoint where and why an error occurred.

For information about the actions that you can perform on server and application logs, see "Using Essbase Server and Application Logs" on page 740. For information on viewing or analyzing logs using Administration Services, see "About Log Viewer" or "About Log Analyzer" in *Oracle Essbase Administration Services Online Help*. For information about specific error messages, see *Oracle Essbase Error Message Reference*.

Contents of the Essbase Server Log

The information in the `ESSBASE.LOG` file can help you assess:

- Who performed an operation
- When an operation was performed
- Errors that occurred when an operation was performed or attempted

Table 77 lists the types of actions logged and the information included in the log message. For information about specific error messages, see *Oracle Essbase Error Message Reference*.

Table 77 Contents of the Essbase Server Log

Type of Action	Information Included
Actions performed at the Essbase Server level, such as logging on Essbase Server or setting security	● User name ● Date and time ● If changing permissions—User and group information and IP address of user ● If logging on—Time and date the user last logged on ● If logging off—Length of time the user was logged on ● If creating or changing an application or database—Request to create or open the artifact; load, connect, and start the application or database to be created; and lock of the artifact to be created or changed ● If creating, changing, or viewing an application or database—Requests to list applications and databases and requests to retrieve access information ● Shutdown or startup requests ● Requests to retrieve and delete the Essbase Server log
Actions performed at the application level, such as viewing application settings or viewing and changing custom-defined macros, custom-defined functions, and substitution variables	● User name ● Date and time ● Requests to retrieve operating system resources, license information, and systemwide configuration ● Retrieving and setting global values ● If altering substitution variables—Request to list and set substitution variables ● If altering custom-defined macros—Request to list and delete custom-defined macros ● Requests to retrieve and delete the application log

Type of Action	Information Included
Actions performed at the database level, such as creating rules files, outlines, reports, or calculation scripts	● User name ● Date and time ● Requests to retrieve client settings, user, and group information ● Requests to list applications and databases ● Requests to retrieve access information ● Requests to lock artifacts ● Returning artifacts

Essbase Server Log Example

The following `essbase.log` example shows entries written when Essbase Server starts. First, the Sample application and the Basic database are loaded. The log includes information about the time the application and database are loaded, the process ID assigned to the application by the operating system, and the startup of the security authentication module.

Note:

You can use the process ID to stop the application improperly if you are unable to perform a normal shutdown. See "Stopping an Application Improperly" on page 703.

```
[Tue Nov 06 07:54:16 2001]Local/ESSBASE0///Info(1054014)
Database Basic loaded

[Tue Nov 06 07:54:16 2001]Local/ESSBASE0///Info(1051061)
Application Sample loaded - connection established

[Tue Nov 06 07:54:16 2001]Local/ESSBASE0///Info(1054027)
Application [Sample] started with process id [1300]

[Tue Nov 06 07:54:18 2001]Local/ESSBASE0///Info(1054014)
Database Basic loaded

[Tue Nov 06 07:54:23 2001]Local/ESSBASE0///Info(1051134)
External Authentication Module: [LDAP] enabled

[Tue Nov 06 07:54:23 2001]Local/ESSBASE0///Info(1051051)
Essbase Server - started
```

The following log shows a single error. The admin user tried to rename an application using a name that already exists on Essbase Server. The log includes information about the user name, the time of the error, and the operation that failed and caused the error.

```
[Tue Nov 06 08:00:04 2001]Local/ESSBASE0///Info(1051001)
Received client request: Rename Application (from user admin)

[Tue Nov 06 08:00:04 2001]Local/ESSBASE0///Error(1051031)
Application Testing already exists

[Tue Nov 06 08:00:04 2001]Local/ESSBASE0///Warning(1051003)
```

```
Error 1051031 processing request [Rename Application] - disconnecting
```

The next log shows a shutdown. The log includes information about the name of the application shutdown and the time of the shutdown.

```
[Tue Nov 06 08:00:46 2001]Local/ESSBASE0///Info(1054005)
Shutting down application Sample

[Tue Nov 06 08:00:52 2001]Local/ESSBASE0///Info(1051052)
Essbase Server - finished
```

Contents of the Application Log

The information in the *appname*.log file can help you assess:

- Who performed a specific operation
- When an operation was performed
- Errors that occurred when an operation was performed or attempted
- Information about dimensions and members to aid in optimization
- The name of an artifact used to execute an operation (such as a calc script or load file used to perform a calculation or a data load) if the artifact resides on an instance of Essbase

Table 78 lists the types of actions logged and the information included in the log message.

Table 78 Contents of the Application Log

Type of Action	Information Included
Actions performed at the database level, such as loading data, clearing data, or calculating data	● User name ● Date and time ● Application and database name and time ● If starting application—Loading Java modules, and reading and writing database and application information ● If loading databases—Information about dimension sizes, dynamic calculation members, blocks, cache sizes, index page size, and I/O information ● If starting databases—Application and database setup information, including reading free space information, writing database parameters, retrieving state information, writing application and database definition information, retrieving database volumes, and writing database mapping ● If loading—Load command, parallel data load information, cells updated, elapsed load time, and the name of the rules file and data file **Note:** Essbase supplies the load rule name only if the client making the request (for example, Administration Services) is at the same release level as Essbase; for example, Release 7.0. ● If calculating—the name of the calculation script used to perform the calculation

Type of Action	Information Included
	● If reporting—the name of the report script
Actions performed at the outline level, such as restructuring	● User name ● Date and time ● Information about dimension sizes, dynamic calculation members, blocks, cache sizes, index page size, I/O information, and restructure elapsed time
Actions performed at the spreadsheet level, such as lock and send	● User name ● Date and time ● Action performed

Example of an Application Log

The following topics show example entries in the application log, including a standard startup and shutdown, and an example of the messages logged when an error occurs.

Example 1: Startup Messages in the Application Log

The following log example shows all the entries written to *appname*.log when Essbase Server starts. The log includes information such as the time the application starts, when application and database information is read and written, when the application is ready for login requests, and when the database is loaded.

```
[Tue Nov 06 08:47:14 2001]Local/Sample///Info(1002035)
Starting Essbase Server - Application [Sample]

[Tue Nov 06 08:47:15 2001]Local/Sample///Info(1200480)
Loaded and initialized JVM module

[Tue Nov 06 08:47:15 2001]Local/Sample///Info(1019008)
Reading Application Definition For [Sample]

[Tue Nov 06 08:47:15 2001]Local/Sample///Info(1019009)
Reading Database Definition For [Basic]

[Tue Nov 06 08:47:15 2001]Local/Sample///Info(1019021)
Reading Database Mapping For [Sample]

[Tue Nov 06 08:47:15 2001]Local/Sample///Info(1019010)
Writing Application Definition For [Sample]

[Tue Nov 06 08:47:15 2001]Local/Sample///Info(1019011)
Writing Database Definition For [Basic]

[Tue Nov 06 08:47:15 2001]Local/Sample///Info(1019022)
Writing Database Mapping For [Sample]

[Tue Nov 06 08:47:15 2001]Local/Sample///Info(1013202)
Waiting for Login Requests
```

```
[Tue Nov 06 08:47:15 2001]Local/Sample///Info(1013205)
Received Command [Load Database]

[Tue Nov 06 08:47:15 2001]Local/Sample///Info(1019018)
Writing Parameters For Database [Basic]

[Tue Nov 06 08:47:15 2001]Local/Sample///Info(1019017)
Reading Parameters For Database [Basic]
```

After Essbase Server starts, it writes information about the dimensions and members in the outline, such as the dimension sizes and dynamic calculation information, to the application log, as shown in the following example:

```
[Tue Nov 06 08:47:15 2001]Local/Sample///Info(1019012)

Reading Outline For Database [Basic]

[Tue Nov 06 08:47:15 2001]Local/Sample///Info(1007043)
Declared Dimension Sizes = [20 17 23 25 5 3 5 3 15 8 6 ]

[Tue Nov 06 08:47:15 2001]Local/Sample///Info(1007042)
Actual Dimension Sizes = [20 14 20 25 4 3 5 3 15 8 5 ]

[Tue Nov 06 08:47:15 2001]Local/Sample///Info(1007125)
The number of Dynamic Calc Non-Store Members = [8 6 0 0 2 ]

[Tue Nov 06 08:47:15 2001]Local/Sample///Info(1007126)
The number of Dynamic Calc Store Members = [0 0 0 0 0 ]
```

Next, Essbase Server writes information about the blocks in the database, including the block size, the number of declared and possible blocks, and the number of blocks needed to perform calculations (you can use this information to estimate the retrieval performance for members of sparse dimensions tagged as Dynamic Calc) to the application log:

```
[Tue Nov 06 08:47:15 2001]Local/Sample///Info(1007127)
The logical block size is [1120]

[Tue Nov 06 08:47:15 2001]Local/Sample///Info(1010008)
Maximum Declared Blocks is [575] with data block size of [1700]

[Tue Nov 06 08:47:15 2001]Local/Sample///Info(1010007)
Maximum Actual Possible Blocks is [500] with data block size of [192]

[Tue Nov 06 08:47:15 2001]Local/Sample///Info(1200481)
Formula for member [Opening Inventory] will be executed in [CELL] mode

[Tue Nov 06 08:47:15 2001]Local/Sample///Info(1012710)
Essbase needs to retrieve [1] Essbase Kernel blocks in order to calculate
the top dynamically calculated block.
```

Next, Essbase Server writes information about the caches set for each database to the application log:

```
[Tue Nov 06 08:47:15 2001]Local/Sample///Info(1012736)
The Dyn.Calc.Cache for database [Basic] can hold a maximum of [2340]
blocks.

[Tue Nov 06 08:47:15 2001]Local/Sample///Info(1012737)
```

```
The Dyn.Calc.Cache for database [Basic], when full, will result in
[allocation from non-Dyn.Calc.Cache memory].

[Tue Nov 06 08:47:15 2001]Local/Sample///Info(1019018)
Writing Parameters For Database [Basic]

[Tue Nov 06 08:47:15 2001]Local/Sample///Info(1019017)
Reading Parameters For Database [Basic]

[Tue Nov 06 08:47:15 2001]Local/Sample///Info(1070013)
Index cache size ==> [1048576] bytes, [1024] index pages.

[Tue Nov 06 08:47:15 2001]Local/Sample///Info(1070014)
Index page size ==> [8192] bytes.

[Tue Nov 06 08:47:15 2001]Local/Sample///Info(1070081)
Using buffered I/O for the index and data files.

[Tue Nov 06 08:47:15 2001]Local/Sample///Info(1070083)
Using waited I/O for the index and data files.

[Tue Nov 06 08:47:15 2001]Local/Sample///Info(1019019)
Reading Data File Free Space Information For Database [Basic]...

[Tue Nov 06 08:47:15 2001]Local/Sample///Info(1006025)
Data cache size ==> [3145728] bytes, [2048] data pages

[Tue Nov 06 08:47:15 2001]Local/Sample///Info(1006026)
Data file cache size ==> [0] bytes, [0] data file pages
```

The final messages logged at startup refer to general database information:

```
[Tue Nov 06 08:47:15 2001]Local/Sample///Info(1013205)
Received Command [Get Database Volumes]

[Tue Nov 06 08:47:15 2001]Local/Sample///Info(1013205)
Received Command [Set Database State]

[Tue Nov 06 08:47:15 2001]Local/Sample///Info(1019018)
Writing Parameters For Database [Basic]

[Tue Nov 06 08:47:15 2001]Local/Sample///Info(1019018)
Writing Parameters For Database [Basic]

[Tue Nov 06 08:47:15 2001]Local/Sample///Info(1013205)
Received Command [Get Database State]

[Tue Nov 06 08:47:15 2001]Local/Sample///Info(1013205)
Received Command [Get Database Info]

[Tue Nov 06 08:47:15 2001]Local/Sample///Info(1013205)
Received Command [SetApplicationState]

[Tue Nov 06 08:47:15 2001]Local/Sample///Info(1019010)
Writing Application Definition For [Sample]

[Tue Nov 06 08:47:15 2001]Local/Sample///Info(1019011)
Writing Database Definition For [Basic]
```

```
[Tue Nov 06 08:47:15 2001]Local/Sample///Info(1019022)
Writing Database Mapping For [Sample]
```

Example 2: Errors in the Application Log

The following example shows a single error. An unknown member was found in the data load file; the presence of an unknown member caused the load to fail. First, you see the request for the data load, then the error message, and, finally, information messages describing the data values changed by the data load and the data load elapsed time.

```
[Tue Nov 06 08:49:52 2001]Local/Sample///Info(1013210)
User [admin] set active on database [Basic]

[Tue Nov 06 08:49:52 2001]Local/Sample/Basic/admin/Info(1013091)
Received Command [DataLoad] from user [admin]

[Tue Nov 06 08:49:52 2001]Local/Sample/Basic/admin/Info(1003040)
Parallel dataload enabled: [1] block prepare threads,
[1] block write threads.

[Tue Nov 06 08:49:52 2001]Local/Sample/Basic/admin/Error(1003000)
Unknown Item [500-10] in Data Load, [0] Records Completed

[Tue Nov 06 08:49:52 2001]Local/Sample/Basic/admin/Warning(1003035)
No data values modified by load of this data file

[Tue Nov 06 08:49:52 2001]Local/Sample/Basic/admin/Info(1003024)
Data Load Elapsed Time : [0.11] seconds

[Tue Nov 06 08:49:52 2001]Local/Sample/Basic/admin/Info(1019018)
Writing Parameters For Database [Basic]
```

Example 3: Shutdown Messages in the Application Log

The following messages are logged when Essbase Server performs a normal shutdown. First, information about the database is retrieved. Then the database is unloaded, free space information is written, and the server shuts down.

```
[Tue Nov 06 08:50:26 2001]Local/Sample///Info(1013214)
Clear Active on User [admin] Instance [1]

[Tue Nov 06 08:50:34 2001]Local/Sample///Info(1013205)
Received Command [Get Database Info]

[Tue Nov 06 08:50:34 2001]Local/Sample///Info(1013205)
Received Command [Get Database State]

[Tue Nov 06 08:50:34 2001]Local/Sample///Info(1013205)
Received Command [Get Database Volumes]

[Tue Nov 06 08:50:34 2001]Local/Sample///Info(1013205)
Received Command [Get Database State]

[Tue Nov 06 08:50:34 2001]Local/Sample///Info(1013205)
Received Command [Get Database Volumes]
```

```
[Tue Nov 06 08:50:34 2001]Local/Sample///Info(1013205)
Received Command [Unload Database]

[Tue Nov 06 08:50:34 2001]Local/Sample///Info(1019018)
Writing Parameters For Database [Basic]

[Tue Nov 06 08:50:34 2001]Local/Sample///Info(1019020)
Writing Free Space Information For Database [Basic]

[Tue Nov 06 08:50:34 2001]Local/Sample///Info(1013207)
RECEIVED SHUTDOWN COMMAND - SERVER TERMINATING
```

Essbase Server and Application Log Message Categories

Table 79 provides error message categories for each error number range that is shown in the first column. When you receive an error message, use this table to identify the Essbase component to which the error is related. See *Oracle Essbase Error Message Reference*.

Table 79 Error Message Categories

Error Message Number Range	Component That Generated the Error
1001000-1001999	Report Writer
1002000-1002999	General server
1003000-1003999	Data load
1004000-1004999	General server
1005000-1005999	Backup, export, or validate
1006000-1006999	Data cache
1007000-1007999	Outline restructure
1008000-1008999	System calls, portable layer, ASD, or Agent
1009000-1009999	Restoring ASCII data
1010000-1010999	Internal (block numbering)
1011000-1011999	Internal (utilities)
1012000-1012999	Calculator
1013000-1013999	Requestor
1014000-1014999	Lock manager
1015000-1015999	Alias table
1016000-1016999	Report Writer
1017000-1017999	Currency

Error Message Number Range	Component That Generated the Error
1018000-1018999	Not currently used
1019000-1019999	Database artifacts
1020000-102999	Spreadsheet extractor
1021000-1021999	Oracle Essbase SQL Interface
1022000-1022999	Security
1023000-1023999	Partitioning
1024000-1024999	Query Extractor
1030000-1030999	API
1040000-1040999	General network
1041000-1041999	Network—Named Pipes
1042000-1042999	Network—TCP
1043000-1049999	Not currently used
1050000-1055999	Agent
1056000-1059999	Not currently used
1060000-1060999	Outline API
106100-1069999	Not currently used
1070000-1070999	Index manager
1071000-1079999	Not currently used
1080000-1080099	Transaction manager
1081000-1089999	Not currently used
1090000-1099999	Rules file processing
1010000-1019999	Not currently used
1100000-1100999	Not currently used
1110000-1119999	Oracle's Hyperion® Web Analysis
1120000-1129999	Grid API
1130000-1139999	Miscellaneous
1140000-1149999	Linked Reporting Objects (LRO)
1150000-1159999	Outline synchronization

Error Message Number Range	Component That Generated the Error
1160000-1169999	Outline change records
1170000-1179999	Attributes
1180000-1189999	Showcase
1190000-1199999	Oracle Essbase Integration Services
1200000-1200999	Calculator framework

Using Essbase Server and Application Logs

The following topics describe the actions you can perform on server and application logs.

For a comprehensive discussion of server and application logs, see "Essbase Server and Application Logs" on page 730. For information about specific error messages, see *Oracle Essbase Error Message Reference*.

You can also view logs using Administration Services. See "About Log Viewer" in *Oracle Essbase Administration Services Online Help*.

Setting the Maximum Log File Size for Essbase Server and Application Logs

You can specify the maximum log file sizes for Essbase Server (Agent) and Application log files in the `essbase.cfg` file using these settings:

- AGTMAXLOGFILESIZE
- APPMAXLOGFILESIZE

The default maximum log file size is 1 GB.

See *Oracle Essbase Technical Reference*.

Setting the Type of Essbase Server Messages Logged

By default, the Essbase Server log, *`HYPERION_HOME`*`/logs/essbase/ESSBASE.LOG`, lists these types of messages:

- Information messages, such as notification of a user action or information about an application or database

  In the following example, the admin user logged out.

  ```
  [Sun Oct 21 16:00:55 2001]Local/ESSBASE0///Info(1051037)
  Logging out user admin, active for 144 minutes
  ```
- Warning messages, such as a notification that an operation was not completed

  Warnings often follow errors. In the following example, the rename operation did not complete because of a previous error in the log.

```
[Fri Nov 02 13:38:14 2001]Local/ESSBASE0///Warning(1051003)
Error 1051031 processing request [Rename Application] - disconnecting
```

- Error messages, such as trying to perform an action that Essbase Server cannot perform

  In the following example, the rename operation failed because the application name already existed.

  ```
  [Fri Nov 02 13:38:14 2001]Local/ESSBASE0///Error(1051031)
  Application Testing already exists
  ```

The following table lists the settings that you specify in the `essbase.cfg` file to determine what types of messages Essbase Server writes to the Essbase Server log. If you change an `essbase.cfg` setting, restart Essbase Server to apply the change.

Setting	Definition	For More Information
AGENTLOGMESSAGELEVEL	An `essbase.cfg` setting that determines whether Essbase Server writes all messages, warning messages, or error messages to the Essbase Server log	*Oracle Essbase Technical Reference*
PORTUSAGELOGINTERVAL	An `essbase.cfg` setting that specifies how often to check the number of ports in use	*Oracle Essbase Technical Reference*

Setting the Type of Application Messages Logged

By default, the application log lists the following types of messages:

- Information messages that detail routine actions that Essbase Server performs

  In the following example, Essbase Server writes the time elapsed during a data load to the application log:

  ```
  [Fri Nov 02 13:04:15 2001]
  Local/Sample/Basic/admin/Info(1003024)
  Data Load Elapsed Time : [3.014] seconds
  ```

- Warning messages that list conditions that are not deemed serious by Essbase Server

  In the following example, Essbase Server writes a statement that no data values were changed during a data load to the application log:

  ```
  [Fri Nov 02 12:43:44 2001]
  Local/Sample/Basic/admin/Warning(1003035)
  No data values modified by load of this data file
  ```

- Error messages that describe errors that occurred while performing the task

  Error messages can range from serious, such as a file not loading correctly, to very serious, such as a disk space error that causes Essbase Server to crash. In the following example, Essbase Server writes a statement to the application log indicating that a data value was encountered before all dimensions in the outline were specified:

  ```
  [Fri Nov 02 12:53:32 2001]
  Local/Sample/Basic/admin/Error(1003007)
  Data Value [678] Encountered Before All Dimensions Selected, [2]
  Records Completed
  ```

Table 80 lists settings that you specify in the `essbase.cfg` file and a command that you can use in a calculation script to determine what types of messages Essbase Server writes to the application log. If you change an `essbase.cfg` setting, restart Essbase Server to apply the change.

Table 80 Setting Messages in the Application Log

Setting	Definition	For More Information
LOGMESSAGELEVEL	An `essbase.cfg` setting that determines whether Essbase Server writes all messages, warning messages, or error messages to the application log	*Oracle Essbase Technical Reference*
TIMINGMESSAGES	An `essbase.cfg` setting that determines whether Essbase Server writes the duration of each spreadsheet and report query to the application log	*Oracle Essbase Technical Reference*
SSLUNKNOWN	An `essbase.cfg` setting that determines whether Essbase Server writes error messages when it encounters an unknown member name during a spreadsheet operation to the application log	*Oracle Essbase Technical Reference*
SET MSG	A calculation script setting that determines whether Essbase Server writes the following items to the application log during the duration of the calculation script: ● A summary of calculation statistics ● Detailed calculation statistics ● All messages, warning messages, error messages, or no messages	*Oracle Essbase Technical Reference*

Viewing the Essbase Server and Application Logs

When you view a log, you are viewing a snapshot. To view an updated version of a log, close and reopen the log. You can view a log from a specific date to the present or to view the entire log.

You must have Administrator permissions to view Essbase Server logs, and you must have at least Application Manager permissions to view application logs.

➤ To view a server or application log, use a tool:

Tool	Topic	Location
Administration Services	Viewing Logs	*Oracle Essbase Administration Services Online Help*
Any text editor		(See the documentation for the text editor.)

Clearing the Essbase Server and Application Logs Immediately

The server and application logs use disk space on the server. Occasionally, you may need to clear entries from a log before it grows too large. Clearing the server log removes all entries in the log but does not remove the server log. Clearing the application log removes all the entries in the application log and deletes the application log. Back up each log before you clear it.

You must have Administrator permissions to clear server and application logs.

➤ To clear a server or application log immediately, use a tool:

Tool	Topic	Location
Administration Services	Deleting Logs	*Oracle Essbase Administration Services Online Help*
MaxL	To clear the Essbase Server log: **alter system clear logfile;** To clear the application log: **alter application *appname* clear logfile;**	*Oracle Essbase Technical Reference*

ESSCMD	To clear the server log:
	DELETELOG "";
	To delete the application log:
	DELETELOG "*appname*";
		Oracle Essbase Technical Reference

Note:

To clear a server or application log each time the server or application restarts, see "Clearing the Essbase Server and Application Logs Upon Restart" on page 743. To conserve space by limiting the items logged in the Essbase Server log or application log, see "Setting the Type of Essbase Server Messages Logged" on page 740 or "Setting the Type of Application Messages Logged" on page 741.

Clearing the Essbase Server and Application Logs Upon Restart

By default, Essbase Server appends messages to the end of the server and application logs. As described in Table 81, you can set Essbase Server to clear the Essbase Server log each time the server is restarted or the application log each time the application is restarted. If you change the `essbase.cfg` file, restart Essbase Server to apply the change.

Table 81 Clearing the Server and Application Logs upon Restart

Setting	Description	For More Information
CLEARLOGFILE	An `essbase.cfg` setting that, when set to TRUE, clears the Essbase Server log each time Essbase Server restarts and the application log each time the application restarts.	*Oracle Essbase Technical Reference*

Note:

To clear a server or application log without restarting the server or application, see "Clearing the Essbase Server and Application Logs Immediately" on page 742. To conserve space by limiting the items logged in the Essbase Server log or application log, see "Setting the Type of Essbase Server Messages Logged" on page 740 or "Setting the Type of Application Messages Logged" on page 741.

Setting Delimiters in the Essbase Server and Application Logs

You can change the symbol used to delimit log entries for server and application logs. Changing the delimiter also affects messages logged in the server console window. By default, Essbase Server uses spaces to delimit fields in a log, as in the following example:

```
[Thu May 10 20:14:46 2001]Local/ESSBASE0///Info(1051051)
Essbase Server - started
```

You can also use tildes, carets, colons, ampersands, or asterisks to delimit the entries in the server and application logs. If possible, choose a delimiter that does not occur frequently in the data, application, or database. The following example shows a log entry delimited by tildes (~):

```
Thu~May~10~20:16:13~2005~Local~ESSBASE0~~~Info~(1051051)~ \\ Oracle
Essbase Server - started
```

Note:

If you set delimiters to anything other than spaces, you can no longer sort the log entries by date in Log Viewer.

The following table lists settings that you specify in the `essbase.cfg` file to determine the delimiters that Essbase Server uses in the server and application logs. If you change an `essbase.cfg` setting, restart Essbase Server to apply the change.

Setting	Description	For More Information
DELIMITEDMSG	An `essbase.cfg` setting that, when set to TRUE, adds a tilde (~) between each field in the server and application logs. To specify a different delimiter, use the DELIMITER setting.	*Oracle Essbase Technical Reference*
DELIMITER	An `essbase.cfg` setting that specifies the delimiter used between each field in the server and application logs. Essbase Server enables you to use the following characters as log delimiters: tilde (~), the default delimiter; caret (^); colon (:); ampersand (&); and asterisk (*). The DELIMTER setting works only when DELIMITEDMSG is set to TRUE.	*Oracle Essbase Technical Reference*

Analyzing Logs with Log Analyzer

You can use Log Analyzer to filter, search, and analyze Essbase Server logs and application logs. Based on filters you choose or create, you can view robust graphical charts for a log. An autorefresh option enables you to monitor log information dynamically.

➤ To use Log Analyzer, see "About Log Analyzer" in *Oracle Essbase Administration Services Online Help*.

Implementing Query Logs

A block storage only feature, query logging enables Essbase administrators to track query patterns of an Essbase database. The query log file tracks all queries performed against the database, regardless of whether the query originated from Oracle Essbase Spreadsheet Add-in or Report Writer. Query logging can track members, generation or level numbers of members belonging to specific generations or levels, and Hybrid Analysis members. Query logging also offers the flexibility to exclude logging of certain dimensions and members belonging to generations or levels. Because the query log file output is an XML document, you can import the log file to any XML-enabled tool to view the log. For information about the query log file structure, refer to `querylog.dtd` in the *`ESSBASEPATH`*`/bin` directory.

To enable query logging, create a query configuration file (distinct from the `essbase.cfg` file) and add to the file the configuration settings that control how query logging is performed.

In the *`ARBORPATH`*`/app/`*`appname`*`/`*`dbname`* directory, create a query log configuration file. The configuration file must be named `dbname.cfg`, where *`dbname`* matches the name of the database. For example, the query log configuration file for Sample.Basic is `basic.cfg`. The output query log file is located at, by default, *`ARBORPATH`*`/app/`*`appname`*`/`*`dbname`*`00001.qlg`.

See the *Oracle Essbase Technical Reference*.

Understanding and Using the Outline Change Log

You can set Essbase Server to create an outline change log, a text file named *`dbname`*`.olg`, that saves outline modification information to a text file. You can review the outline change log anytime to see the changes that have been made to an outline since the log was created. This information helps you to roll back an outline to a previous version.

The outline change log is located in the following directory:

`HYPERION_HOME``/logs/essbase/app/`*`appname`*`/`*`dbname`*

The following topics describe the outline change log and the actions that you can perform on it.

Understanding the Contents of the Outline Change Log

Table 82 lists the types of actions written to the outline change log and the information included in the log message.

Table 82 Outline Change Log Contents

Type of Change	Information Included
Add a dimension	● Name of dimension

Type of Change	Information Included
	● Type of dimension (dense or sparse) ● Dimension tag (if any) ● Name of left sibling of dimension (if any) ● Level number of dimension ● Generation number of dimension
Delete a dimension	Name of dimension
Update a dimension	● Name of dimension ● Dimension tag ● Type of dimension (dense or sparse) ● Level name changes (if applicable) ● Generation name changes (if applicable)
Rename a dimension	● Old name of dimension ● New name of dimension
Move a dimension to a new position	● Name of dimension ● Old location, including left sibling of dimension ● New location, including left sibling of dimension
Add a member to a dimension	● Name of new member or members ● Unary calculation symbol for member ● Level number of member ● Generation number of member ● Status of member (Store, Share) ● Member alias (if applicable) ● Account type of member (if applicable) ● UDAs of member (if applicable) ● Calculation formula for member (if applicable)
Update a member of a dimension	● Name of member updated ● Member properties that were updated
Rename a member of a dimension	● Old name of member ● New name of member
Move a member of a dimension to a new position	● Name of member moved ● New location ● Names of parent and left sibling in new location

The outline change log program reads outline information from left to right. If you are looking at an outline, the *left sibling* is the sibling directly above (to the left of) the newly added dimension or member. This rule does not apply if the immediately preceding dimension or member is a *parent*. If a newly added (or moved) member is the first child of its parent, or if the member is the first dimension in the outline, the outline change log identifies the old location of the dimension or member as None.

Reviewing an Example of an Outline Change Log

When a user makes and saves changes to an outline, Essbase writes the change information into the outline change log as a group of entries. Each group of entries begins and ends with identifying information so that you can easily identify each revision, from newest to oldest.

The following change log shows one record, indicating that since the outline change log was started, the outline was modified once. First, the outline change log lists the beginning of the change, including the application and database changed; the time of the change; and the user making the change. Next, the outline change log lists the change—a member named 100-50 was added to 100 member of the Product dimension. Finally, the outline log lists the end of the change, including the application and database changed; the time of the change; and the user making the change.

```
[Begin Outline Change for Sample/Basic, Sat Nov 03 12:49:31 2001, By admin]
Number of member changes for dimension "Product" : 1

Added new member "100-50" to "100" :
Left sibling - "100-40"
Status - Store Data
Added alias "Cherry Cola" to alias table "Default"
Unary calc symbol - Add
Level Number - 0
Generation Number - 3
[End Outline Change for Sample/Basic, Sat Nov 03 12:49:32 2001, By admin]
```

Creating Outline Change Logs

By default, Essbase Server does not create an outline change log. To create one, use the OUTLINECHANGELOG TRUE configuration setting in `essbase.cfg`. See the *Oracle Essbase Technical Reference*.

Note:

During a restructure, Essbase holds outline change information in memory until all updates have been made to the outline change log. Turning on the outline change log may, therefore, affect restructure performance. See “Conditions Affecting Database Restructuring” on page 843.

Viewing Outline Change Logs

You can view the outline change log by opening the log file in a text editor.

Setting the Size of the Outline Change Log

The default size for the outline change log (`dbname.olg`) is 64,000 bytes. To change the maximum size, use the OUTLINECHANGELOGFILESIZE configuration setting in the `essbase.cfg` file. See the *Oracle Essbase Technical Reference*.

When the outline change log reaches the maximum size, Essbase copies the contents of the `.olg` file to a new file, changing the extension to `.olb`. For example, when `basic.olg` reaches the maximum size, the contents are copied to `basic.olb`.

After the copy operation, the `.olg` file is cleared and Essbase writes new log entries to it.

Each time the `.olg` files reaches its maximum file size, Essbase overwrites the existing `.olb` file with the current contents of the `.olg`. Therefore, be sure to retrieve information from the `.olb` file before it is overwritten.

The default, minimum, and maximum file sizes for the `.olg` file are also automatically applied to the `.olb` file. For example, if you change the maximum size of the outline change log to 2 MB, the `.olb` file is automatically set to the same maximum size.

Understanding and Using Exception Logs

When an Essbase Server, an application, or a database shuts down abnormally, Essbase Server sometimes creates an exception log as a text file named `log0000n.xcp`. The following topics describe the server, application, and database exception logs and the actions that you can perform on them.

Understanding the Contents of Exception Logs

If an Essbase Server, an application, or a database shuts down abnormally and cannot restart, Essbase Server generates an exception log to help troubleshoot the problem. The location of the exception log depends on which component shut down abnormally and the amount of information that Essbase Server had available at the time. Table 84 on page 753 describes the location of the exception log for each type of abnormal shutdown.

Table 83 lists the sections of the exception log and the information included in each section. If Essbase Server could not retrieve all the information before the shutdown finished, some of the later sections may be blank.

Table 83 Contents of the Exception Log (`log00001.xcp`)

Section of Log	Information Included
General information	● Date and time ● Application and database name ● Location of exception log ● Process type Use this information to determine which component shut down abnormally and when it shut down.

Section of Log	Information Included
Machine registers and stack information	● General registers ● Floating point registers ● Hex registers ● Stack trace ● Stack dump Oracle Support can examine this section of the log to help determine why an abnormal shutdown may have occurred.
Application-wide configuration	● Server and application name ● Elapsed application time; that is, how long the application was running ● List of modules Use this information to determine whether the application shut down quickly and that all modules are correct. More information about modules is in the systemwide configuration section of the exception log.
Operating system resources	● System date and time ● Elapsed operating system time; that is, how long the operating system was running ● Resource information, including CPU type, memory information, swap information, and drive information Use this information to see if it is an operating system problem, such as a lack of memory.
System-wide configuration	● Elapsed Essbase time; that is, how long Essbase was running ● Essbase release information ● Network information ● Environment variables ● Module information, including module name and release Use this information to ensure that the release is the same for Essbase and each module, and that environment variables are set correctly.
`essbase.cfg` values	Values of all settings in the `essbase.cfg` file. Use this information to make sure that Essbase Server is configured correctly.
License information	● Serial number and license expiration date ● Number of ports purchased and ports in use ● Essbase options enabled ● Other Oracle Hyperion products enabled Use this information to ensure that the correct options of Essbase are installed and that you have purchased enough ports.
Client request activity	● Server name ● Application name ● Thread information, including the number of threads ● Request information

Section of Log	Information Included
	● Detailed information about each thread, including the action it is performing, the database, user name, start time, and end time Use this information to determine how heavy the load on the server was, based on client requests.
File information	● Page file size ● Index file size Use this information to determine whether the page file or the index is too large.
Database information	Use this information to ensure that the database is set up correctly.
Database statistics	Use this information to view dimension information and to see characteristics of data blocks in the database.

Reviewing an Example of an Exception Log

The following example is of an exception log. The first section of the log lists general information about the application and database. In this example, the Essbase Server shut down:

```
----- Exception Error Log Begin -----

Current Date & Time:    Sat Nov 24 13:25:13 2001
Process Type:           Server
Application Name:       Sample
Database Name:          Basic
Exception Log File:     C:\HYPERION\ESSBASE\log00001.xcp
Current Thread Id:      1116
Exception Code:         0xC0000005=Access Violation
Exception Flags:        0x00000000=Continuable
Exception Address:      0x002D2249
Exception Parameters:   2
Exception Parameter 0: 0x00000000=Read Violation
Exception Parameter 1: 0x0000220A (Virtual Address)
```

The next section of the log lists register and stack trace information. Oracle Support can examine this section of the log to assist in determining why an abnormal shutdown occurred.

```
----- Machine Registers -----

General Registers:
   EAX=0x00000000  EBX=0x01358008  ECX=0x00002200

Control Registers:
   CS =0x0000001B  EIP=0x002D2249  Flg=0x00010202

Segment Registers:
   DS =0x00000023  ES =0x00000023  FS =0x00000038

Floating Point Registers:
   CWD=0xFFFF027F  SWD=0xFFFF0000  TWD=0xFFFFFFFF

Register Area (Hex):
   00 00 00 00 00 00 00 00 00 00

...continued hexadecimal listings...

Debug Registers:
   DR0=0x2F75C73B  DR1=0x75E07D39  DR2=0x1475FF16
   DR3=0x00000000  DR6=0x0000E00B  DR7=0x00000000
----- Stack -----

Stack Trace:
    0: 0x002D2249
    1: 0x002D202D
...continued stack trace listings...

Stack Dump (Hex):
   (Stack dump truncated from 1397 to 1024 bytes.)
   0x0012EA2C:  00000000 01358008 01358008 FFFFFFFF
   0x0012EA3C:  00002200 002D728C 0012EC6C 002D202D
...continued stack dump listings...
```

The following section of the log lists application information:

```
----- Application-Wide Configuration -----

Server Name:           ASPEN
Application Name:      Sample
Elapsed App Time:      00:00:01:28
Module Count:          6
Module  0:             0x00241000 = C:\HYPERION\ESSBASE\BIN\ESSSEC.DLL
Module  1:             0x002C1000 = C:\HYPERION\ESSBASE\BIN\ESSNET.DLL
...continued module listings...
```

The following section of the log lists operating system information. You can determine how much memory is available, how much swap space is used, and how much memory is available on each drive:

```
----- Operating System Resources -----

System Date & Time:    Sat Nov 24 13:25:13 2001
Elapsed OS Time:       02:03:03:10
OS Name & Version:     Windows NT 5.00
CPU Count:             1
CPU Type:              Pentium
Total Physical Memory: 327024 KB (334872576)
Free Physical Memory:  155760 KB (159498240)
Used Physical Memory:  171264 KB (175374336)
Swap Flags:
   Enabled:            Y
   Disabled:           N
   File Found:         Y
   Denied:             N
Swap file(s):          C:\pagefile.sys
Total Swap Space:      467192 KB (478404608)
Free Swap Space:       421528 KB (431644672)
Used Swap Space:       45664 KB (46759936)
Total Drives:          5
Current Drive:         3
Drive  1:
   Drive Name:         C
   Volume Label:
   Drive Type:         Fixed
   File System:        NTFS
   Total Drive Space:  11778448 KB
  Free Drive Space:   8592548 KB
   Used Drive Space:   3185900 KB
...continued drive listings...
```

The following section of the log lists system configuration information, such as paths or `essbase.cfg` settings:

```
----- System-Wide Configuration -----

Elapsed Essbase Time:  00:00:01:33
Essbase Version:       6.2.0
Essbase Description:   Ess62P0B128
Network Type:          Windows Sockets
Environment Variable:  ARBORPATH = C:\HYPERION\ESSBASE
Environment Variable:  ARBORMSGPATH = C:\HYPERION\ESSBASE\bin
Module Count:          13
Module  0:
   Module Name:        C:\HYPERION\ESSBASE\BIN\ESSUTL.DLL
   Module Version:     6.2.0.1
   Module Description: Ess62P0B128.1
   Module Use Count:   5
...continued module listings...

----- ESSBASE.CFG Configuration Values -----

Configuration Value:   JvmModuleLocation = C:\Hyperion\Essbase\java\jre13
\bin\hotspot\jvm.dll
Configuration Value:   AuthenticationModule = LDAP essldap.dll x
Configuration Value:   OUTLINECHANGELOG = TRUE
```

The following section of the log lists license information (such as a serial number), Essbase options (such as ports purchased) and Oracle Hyperion products purchased:

```
----- License Information -----

Serial Number:         xxx
License Expiry Date:
Port Count:            10
Ports In Use Count:    0
Limited Use Version:   N
Read-Only SS:          N

...continued Essbase options and Hyperion product listings...
```

The following section of the log lists client activity, such as using Administration Services to view databases or using the Spreadsheet Add-in to view databases:

```
----- Client Request Activity -----

Server Name:           ASPEN
Application Name:      Sample
Total Request Threads: 5
Avail Request Threads: 6
```

```
Total Requests:         56
Average Requests:       48.000000
Weighted Average:       7.440000
Statistics Per Minute:
   Current Requests:    48
   Minimum Requests:    48.000000
   Maximum Requests:    48.000000
Thread Count:           5
Thread Id 1444:
   Request Name:         List Objects
   Database Name:        Basic
   User Name:            admin
   Start Time:           Sat Nov 24 13:24:37 2001
   End Time:             Sat Nov 24 13:24:37 2001
...continued thread listings...

----- Exception Error Log End -----
```

Viewing Exception Logs

You can view the exception change log by opening it in any text editor. The location of the exception log depends on which component shut down abnormally and the amount of information that Essbase Server had available at the time.

Table 84 describes the location of the exception log.

Table 84 Location of the Exception Log

Component That Shut Down	Location of the Exception Log
Essbase Server	The log is in the *ESSBASEPATH* directory; for example: `Hyperion/products/Essbase/EssbaseServer/log00001.xcp`
Application	If the application name is unknown, the log is in the *ARBORPATH*/`app` directory. For example: `Hyperion/products/Essbase/EssbaseServer/app/log00001.xcp` If the application name is known, the log is in the application directory. For example, if the Sample application shut down abnormally: `Hyperion/products/Essbase/EssbaseServer/app/Sample/log00001.xcp`
Database	If the database name is unknown, the log is in the application directory; for example, if the Basic database shut down abnormally: `Hyperion/products/Essbase/EssbaseServer/app/sample/log00001.xcp` If the database name is known, the log is in the database directory; for example, if the Basic database shut down abnormally: `Hyperion/products/Essbase/EssbaseServer/app/sample/basic/log00001.xcp`

Overwriting or Retaining Existing Logs

By default, Essbase Server creates one or more exception logs each time the server shuts down abnormally. Subsequent exception logs are numbered sequentially; for example, if `log00001.xcp` exists, the next log is named `log00002.xcp`.

While you can use the EXCEPTIONLOGOVERWRITE TRUE configuration setting in `essbase.cfg` to overwrite the existing exception log instead of creating a new log, Oracle recommends that you use the default setting of FALSE. An abnormal shutdown may create multiple exception logs, and the first log created during the shutdown is often the most descriptive. See *Oracle Essbase Technical Reference*.

Understanding and Using Dimension Build and Data Load Error Logs

Essbase Server writes errors that occur during a dimension build or data load in error logs. The log that Essbase Server chooses for errors depends on the operation that you perform, such as a dimension build or a data load, and how you perform it, such as using Administration Services, or MaxL. The following topics describe the location of dimension build and data load errors and the actions that you can perform on dimension build and data load error logs.

Understanding and Viewing Dimension Build and Data Load Error Logs

The `dataload.err` log contains errors that occurred during a dimension build or a data load. The logs also contain the records that failed to load. After you fix the errors, you can reload the logs. See "Loading Dimension Build and Data Load Error Logs" on page 755.

Essbase Server writes errors that occur during a dimension build or data load to the following error logs:

Table 85 Location of Dimension Build and Data Load Error Logs

Operation	Location of Error Log
Dimension build	*`ESSBASEPATH`*`/client/dataload.err`
Data load with a rules file	*`ESSBASEPATH`*`/client/dataload.err`
Data load without a rules file	*`ARBORPATH`*`/app/`*`appname`*`/`*`appname`*`.log`

Note:

If the data load or dimension build fails and there is no error log, see "Checking Error Logs" on page 303 for a description of possible causes.

To set the location and file name of dimension build and data load error logs, see "Performing a Data Load or Dimension Build" in *Oracle Essbase Administration Services Online Help*.

➤ To view the dimension build and data load error logs, open them in any text editor.

Reviewing an Example of a Dimension Build and Data Load Error Log

The following data log entry indicates that because the 500 member did not exist in the outline, no data was loaded to it.

```
\\ Member 500 Not Found In Database
500   500-10   500-10-10
```

To resolve this problem, you can perform either of the following actions and restart the load:

- Perform a dimension build to create the missing member. See "Performing Data Loads or Dimension Builds" on page 297.
- Add the missing member in Outline Editor. See "Adding Dimensions and Members to an Outline" on page 129.

The following dimension build log entry indicates that the 600-20 member is not the parent of the 600 member. Ensure that you use the correct rules file with the correct text file. The record looks like it is for a level (bottom-up) build, but the error message indicates that Essbase Server is trying to perform a generation (top-down) build. After you correct the problem, restart the dimension build.

```
\\Record #2 - Incorrect Parent [600-20] For Member [600] (3307)
600-20-10   600-20   600
```

Setting the Maximum Number of Errors

The default size of the `dataload.err` files is 1,000 records. When the log reaches this size, Essbase Server no longer writes errors to the log. The dimension build or data load, however, continues. You can set Essbase Server to write 1 to 65,000 records in the `dataload.err` files. If you change the `essbase.cfg` file, restart Essbase Server to apply the change.

Table 86 Setting the Number of Error Records

Setting	Description	For More Information
DATAERRORLIMIT	An `essbase.cfg` setting that determines the number of records logged in the `dataload.err` log	*Oracle Essbase Technical Reference*

Loading Dimension Build and Data Load Error Logs

If the dimension build or data load fails, you must determine whether the Isolation Level transaction setting is Committed or Uncommitted. If the transaction setting is Committed, you must restart the data load from the beginning. If it is Uncommitted, you can load only the records that failed by loading the error log. Reloading only the failed records is much faster than reloading every record.

➤ To reload the error log:

1 **If you load from the server, change the file extension from `.err` to `.txt`. For example, change the `dataload.err` file to `dataload.txt`.**

   If you load from the client, you can leave the `.err` extension.

2 **Fix the problem that caused the dimension build or data load to fail. Fixing the problem might involve changing the outline, the text in the error log, or the rules file.**

   Check whether the following conditions are true:

   - Can you validate the rules file?

     See "Setting Dimension Build Operational Instructions" on page 280.

   - Is the data source available?

     See "Verifying that the Data Source Is Available" on page 302.

   - Is the server available?

     See "Verifying that Essbase Server Is Available" on page 302.

3 **Load the error log using the appropriate rules file.**

   See Chapter 16, "Understanding Data Loading and Dimension Building."

46 Managing Database Settings

In This Chapter

Understanding the Essbase Server Kernel

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

The kernel has the following functions:

- Handles disk storage and caching of Essbase files
- Handles data retrieval
- Handles data updates
- Controls input/output functions related to Essbase
- Consolidates free space for reuse
- Manages concurrent operations
- Recovers databases after a server crash
- Issues locks
- Manages transactions

The rest of this section explains the two available access modes, and describes how to set the access modes.

Note:

For information about fatal errors in the Essbase Server Kernel, see "Understanding Fatal Error Handling" on page 1015.

Understanding Buffered I/O and Direct I/O

The Essbase Kernel uses buffered I/O (input/output) by default, but direct I/O is available on most of the operating systems and file systems that Essbase supports. For a list of the supported platforms, see the *Oracle Hyperion Enterprise Performance Management System Installation Start Here*.

Buffered I/O uses the file system buffer cache.

Direct I/O bypasses the file system buffer cache and is able to perform asynchronous, overlapped I/Os. The following benefits are provided:

- Faster response time. A user waits less time for Essbase to return data.
- Scalability and predictability. Essbase lets you customize the optimal cache sizes for its databases.

If you set a database to use direct I/O, Essbase attempts to use direct I/O the next time the database is started. If direct I/O is not available on your platform when the database is started, Essbase uses buffered I/O, which is the default. However, Essbase will store the I/O access mode selection in the security file and attempts to use that I/O access mode each time the database is started.

Note:

Cache memory locking can only be used if direct I/O is used. You also must use direct I/O if you want to use an operating system's no-wait (asynchronous) I/O.

Viewing the I/O Access Mode

Buffered I/O is the default for all databases.

➤ To view which I/O access mode a database is currently using or is currently set to, use a tool:

Tool	Topic	Location
Administration Services	Selecting an I/O Access Mode	*Oracle Essbase Administration Services Online Help*
MaxL	**display database**	*Oracle Essbase Technical Reference*
ESSCMD	GETDBINFO	*Oracle Essbase Technical Reference*

Setting the I/O Access Mode

➤ To use direct I/O instead of the default buffered I/O for any database, use a tool:

Tool	Topic	Location
Administration Services	Selecting an I/O Access Mode	*Oracle Essbase Administration Services Online Help*

Tool	Topic	Location
MaxL	**alter database**	*Oracle Essbase Technical Reference*
ESSCMD	SETDBSTATEITEM	*Oracle Essbase Technical Reference*

You may also need to increase the size of some caches. See "Sizing Caches" on page 826.

Understanding Kernel Components

The kernel contains components that control all aspects of retrieving and storing data:

- The Index Manager finds and tracks the location of requested data. See "Index Manager" on page 759.
- The Allocation Manager, part of the Index Manager, allocates space and manages some file operations. See "Allocation Manager" on page 760.
- The Data Block Manager retrieves the data pointed to by the index and stores the data. See "Data Block Manager" on page 760.
- The LRO Manager handles retrieval and storage of LROs. See "LRO Manager" on page 761.
- The Lock Manager handles the locking of data blocks to regulate concurrent data access. See "Lock Manager" on page 761.
- The Transaction Manager tracks transactions and handles internal commit and abort operations. See "Transaction Manager" on page 761.

Index Manager

The Index Manager manages the database index and provides a fast way to look up Essbase data blocks. The Index Manager determines which portions of the database index to cache in the index cache, and manages the index cache.

The Index Manager controls five components, described in the following table:

Component	Description
Index	The method that Essbase uses to locate and retrieve data. The term index also refers to the index file.
Index file	File that Essbase uses to store data retrieval information. It resides on disk and contains index pages. Essbase names index files incrementally on each disk volume, using the naming convention `ess`*`xxxxx`*`.ind`, where *`xxxxx`* is a number. The first index file on each disk volume is named `ess00001.ind`.
Index page	A subdivision of an index file that contains index entries that point to data blocks.
Index entry	A pointer to a data block. An index entry exists for every intersection of sparse dimensions.
Index cache	A buffer in memory that holds index pages.

The term *index* refers to all index files for a single database. The index can span multiple volumes, and multiple index files can reside on a single volume. Use the disk volumes setting to specify disk spanning parameters. For information on setting the index cache size, see "Sizing the Index Cache" on page 827. For information about allocating storage space with the disk volumes setting, see "Specifying Disk Volumes" on page 768.

Allocation Manager

Allocation Manager, part of the Index Manager, performs these tasks:

- Creation and extension of index and data files on disk
- File open and close operations
- Designation of which volume to use for a new file
- Sequence of volume use

When one of these tasks must be performed, the Allocation Manager uses this process to allocate space:

1. It attempts to use free space in an existing file.
2. If not enough free space is available, it attempts to expand an existing file.
3. If not enough free space is available in existing files, it creates a file on the current volume.
4. If it cannot expand a file or create a file on the specified volume, it attempts to use the next specified volume.
5. If all specified volumes are full, an error message is displayed, and the transaction is aborted.

The Allocation Manager allocates space for index and data files based on the database settings for storage.

➤ To check current values and set new values, use a tool:

Tool	Topic	Location
Administration Services	Setting Database Properties	*Oracle Essbase Administration Services Online Help*
MaxL	**alter database**	*Oracle Essbase Technical Reference*
ESSCMD	SETDBSTATEITEM 23	*Oracle Essbase Technical Reference*

See "Specifying Disk Volumes" on page 768.

For information on how Essbase stores data, see "Storage Allocation" on page 767.

Data Block Manager

The Data Block Manager brings data blocks into memory, writes them out to data files, handles data compression, and writes data files to disk. The Data Block Manager controls four components. The following table describes each component:

Component	Description
Data file	A file that contains data blocks. Essbase generates the data file upon data load and stores it on disk. Essbase names data files incrementally—`ess`*`xxxxx`*`.pag`, where *`xxxxx`* is a number, starting with 00001.
Data block	The primary storage unit within Essbase. A data block is a multidimensional array that represents cells of the dense dimensions for a given intersection of sparse dimensions.
Data cache	A buffer in memory that holds uncompressed data blocks.
Data file cache	A buffer in memory that holds compressed data files (`.pag`).

The size of the data file cache determines how much of the data within the data files can fit into memory at one time. The data cache size and the data block size determine how many data blocks can fit into memory at one time. Data files for a single database can span multiple volumes; multiple databases can reside on the same volume. For information on setting the data file cache size and data cache size, see "Sizing the Data File Cache" on page 828 and "Sizing the Data Cache" on page 829. For information about allocating storage space with the disk volumes setting, see "Specifying Disk Volumes" on page 768.

LRO Manager

LROs enable you to associate objects, such as flat files, with data cells. Using the Spreadsheet Add-in, users can create and store LRO files, with an `.lro` extension.

LRO files are stored in the database directory (*`ARBORPATH`*`/app/`*`appname`*`/`*`dbname`*; for example, `app/Sample/Basic`).

Essbase stores information about LROs in an LRO catalog. Each catalog resides in its own Essbase index page and coexists in an index file with other, non-LRO Essbase index pages.

See Chapter 11, "Linking Objects to Essbase Data" and the *Oracle Essbase Spreadsheet Add-in User's Guide*.

Lock Manager

The Lock Manager issues locks on data blocks, which in turn controls concurrent access to data.

The *committed access* and *uncommitted access* isolation levels use different locking schemes. For more information on isolation levels and locking, see Chapter 48, "Ensuring Data Integrity."

Transaction Manager

The Transaction Manager controls transactions and commit operations and manages database recovery.

Essbase commits data automatically. Commits are triggered by transactions that modify data—data loading, calculating, restructuring, and spreadsheet lock and send operations.

How Essbase commits data depends on whether the transaction isolation level is set to committed or uncommitted access (the default). See "Committed Access" on page 781 and "Uncommitted Access" on page 783.

The Transaction Manager maintains a transaction control table, *dbname*.tct, to track transactions.

For information about commit operations and recovery, see "Recovering from a Crashed Database" on page 789.

Understanding Kernel Startup

The sequence of events during kernel startup:

1. After the Essbase Server starts, a user connects to it from a client.
2. The user starts a database.
3. Essbase loads the database.
4. The Essbase Agent passes database settings to the server.
5. The kernel begins its initialization process.
6. The kernel starts its components—the Index Manager, Lock Manager, LRO Manager, Data Block Manager, and Transaction Manager.

If it encounters an error during start up, the Essbase Kernel shuts itself down.

Understanding the Precedence of Database Settings

Essbase provides default values for some database storage settings in the essbase.cfg file. You can leave the default settings or change their values in two places:

- You can define storage settings for all databases on the Essbase Server by changing values in the essbase.cfg file.
- You can define storage settings for a single database by using any of the methods specified in "Specifying and Changing Database Settings" on page 764.

Changes made for an individual database permanently override essbase.cfg settings and Essbase defaults for the relevant database until they are changed or withdrawn.

If you use MaxL or Administration Services Console to change settings at the database level, the changes become effective at different times, as shown in Table 87:

Table 87   When Database-Level Storage Settings Become Effective

Setting	When setting becomes effective
● Index cache ● Data file cache ● Data cache	After you stop and restart a database

Setting	When setting becomes effective
● Cache memory locking ● Disk volume	
Isolation level parameters, concurrency parameters	The first time after setting these values that there are no active transactions
All other settings	Immediately

If you manually change these database settings in `essbase.cfg`, you must stop and restart the relevant application to make them effective.

Note:

The size of index pages is fixed at 8 KB to reduce input-output overhead, as well as to simplify database migration.

Understanding How Essbase Reads Settings

Essbase reads `essbase.cfg` when you start Essbase Server and applies settings to the databases that you created using the methods described in "Specifying and Changing Database Settings" on page 764.

Database settings that you specify using Administration Services, ESSCMD, or MaxL always override `essbase.cfg` settings, even if you change a setting in `essbase.cfg` after you have applied a setting for a database. Only removing a setting triggers Essbase to use `essbase.cfg`, and then only after Essbase Server is restarted.

Viewing Most-Recently Entered Settings

➤ To view the most-recently entered settings, use a tool:

Tool	Topic	Location
Administration Services	Setting Database Properties	*Oracle Essbase Administration Services Online Help*
MaxL	**display database**	*Oracle Essbase Technical Reference*
ESSCMD	GETDBSTATE GETDBINFO	*Oracle Essbase Technical Reference*

Customizing Database Settings

You can customize settings for each database on Essbase Server. The information in this section helps you understand what each setting controls, how to specify settings, and lists examples. For a table of performance-related settings, see Chapter 51, "Improving Essbase Performance."

Note:

Configure settings that are applied to an entire Essbase Server in `essbase.cfg`. For more information, see the *Oracle Essbase Technical Reference*.

You can customize the following major database settings:

Table 88 Major Kernel Settings

Setting	More Information
Index cache size	"Sizing the Index Cache" on page 827
Data file cache size	"Sizing the Data File Cache" on page 828
Data cache size	"Sizing the Data Cache" on page 829
Cache memory locking	"Deciding Whether to Use Cache Memory Locking" on page 826
Disk volumes	"Storage Allocation" on page 767
Data compression	"Data Compression" on page 772
Isolation level	"Understanding Isolation Levels" on page 779

The following sections describe how to change kernel settings and list examples.

Specifying and Changing Database Settings

Before you change database settings, review information about precedence of the settings as changed in different parts of Essbase, and how Essbase reads those settings:

- "Understanding Kernel Startup" on page 762
- "Understanding How Essbase Reads Settings" on page 763

➤ To specify most database settings, use a tool:

Tool	Topic	Location
Administration Services	Setting Database Properties	*Oracle Essbase Administration Services Online Help*
MaxL	**alter database**	*Oracle Essbase Technical Reference*
ESSCMD	SETDBSTATEITEM SETDBSTATE	*Oracle Essbase Technical Reference*

These methods provide different ways to change the same database settings. In rare cases, you may want to use `essbase.cfg` to specify settings.

Caution!

In previous versions of Essbase, you can specify many database settings in the `essbase.cfg` file on Essbase Server. In Version 5.*x* and later, Essbase overrides most of the `.cfg` settings. For an explanation of how newer versions of Essbase handle settings, see "Understanding the Precedence of Database Settings" on page 762 and "Understanding How Essbase Reads Settings" on page 763.

Using alter database in MaxL

Issue a separate **alter database** statement for each database setting you want to change. For example, the following MaxL script logs on to Essbase, changes three database settings, and logs off:

```
login admin identified by secretword;
alter database sample.basic enable committed_mode;
alter database sample.basic set lock_timeout immediate;
alter database sample.basic disable create_blocks;
logout;
```

Note:

Terminate each MaxL statement with a semicolon when issuing them using the MaxL Shell; however, if MaxL statements are embedded in Perl scripts, do *not* use the semicolon statement terminator.

You can use MaxL to write batch scripts that automate database setting changes. See the *MaxL Language Reference*, located in the *Oracle Essbase Technical Reference*.

Using SETDBSTATEITEM in ESSCMD

For simple items, specify the command, item number representing the parameter, application, database, and value for the parameter:

```
SETDBSTATEITEM 2 "SAMPLE" "BASIC" "Y";
```

For parameters that require multiple values, such as Isolation Level (item 18), specify multiple values; in this case, all the values after "BASIC":

```
SETDBSTATEITEM 18 "SAMPLE" "BASIC" "1" "Y" "-1";
```

If you do not know the parameter number, omit it, and Essbase lists all parameters and their corresponding numbers. Essbase also prompts you for a database and an application name.

Use a separate SETDBSTATEITEM command for each parameter; you cannot string parameter numbers together on the same line.

See the *Oracle Essbase Technical Reference* for information about the parameters for the SETDBSTATE and SETDBSTATEITEM commands.

Note:

SETDBSTATEITEM or SETDBSTATE affects only the specified database.

You can include SETDBSTATEITEM (or SETDBSTATE) in batch scripts. For a comprehensive discussion of batch processing, see "Using Script and Batch Files for Batch Processing" on page 1052. For information on specific ESSCMD syntax, see the *Oracle Essbase Technical Reference*.

47 Allocating Storage and Compressing Data

In This Chapter

Storage Allocation

Essbase uses a data file to store data blocks. By default, a data file is located in its associated database folder. Data files follow the naming convention `ess`*`n`*`.pag`, where *n* is greater than or equal to one and less than or equal to 65,535.

Essbase uses an index file to store the index for a database. By default, an index file is located in its associated database folder. Index files follow the naming convention `ess`*`n`*`.ind`, where *n* is greater than or equal to 1 and less than or equal to 65,535.

Essbase automatically allocates storage for data and index files. You can use disk volumes to control how storage is allocated for these files.

➤ To specify disk volumes so that you control how storage is allocated:

1 **Verify how much space Essbase uses to store index and data files. See "Checking Index and Data File Sizes" on page 767 for information about how to check sizes.**

2 **Choose a technique to control storage:**

- Specify which volumes (drives) Essbase uses to store these files. See "Specifying Disk Volumes" on page 768.
- Install Essbase on one volume and store files on another.

Checking Index and Data File Sizes

➤ To view index file (`.ind` file) and data file (`.pag` file) names, counts, sizes, and totals, and to determine whether each file is open in Essbase, use a tool:

Tool	Topic	Location
Administration Services	Checking Index and Data File Sizes	*Oracle Essbase Administration Services Online Help*

Tool	Topic	Location
ESSCMD	LISTFILES	*Oracle Essbase Technical Reference*

Note:

The file size information that is provided by Windows for index and data files that reside on NTFS volumes may not be accurate. The file size information provided by Administration Services and by LISTFILES is accurate.

Specifying Disk Volumes

Use disk volumes to specify where you want to store Essbase index files (`ess`*`n`*`.ind`) and data files (`ess`*`n`*`.pag`). If you do not use the disk volumes setting, Essbase stores data only on the volume where the *`ARBORPATH`* directory resides. If the *`ARBORPATH`* variable is not set, Essbase stores data only on the volume where the server was started.

Note:

For information about how to check the size of the index and data files, see "Checking Index and Data File Sizes" on page 767.

You can specify disk volumes using Administration Services, MaxL, or ESSCMD. When you use disk volumes, Essbase provides the following options for each:

- Volume name
- Maximum space to use on the volume (called Partition Size in Administration Services and Volume Size in ESSCMD)
- File type. You can specify index files, data files, or both. The default is index and data files on the same volume.
- Maximum file size. The default and recommended value is 2,097,152 KB (2 GB). When Essbase reaches the maximum file size, it creates a file and names it incrementally. For example, when `ess00001.ind` is filled to maximum size, Essbase creates `ess00002.ind`.

  Caution!

  If you specify a volume name but not a volume size, Essbase uses all available space on the volume.

Essbase creates data files and index files in these situations:

- If the total sizes of all files reach the maximum size that you specified in the disk volumes setting. By default, the total is the sum of all index and data file sizes. If you specify Index as the file type, the total refers to the sum of all index files on a volume. If you specify Data as the file type, the total refers to the sum of all data files on a volume.

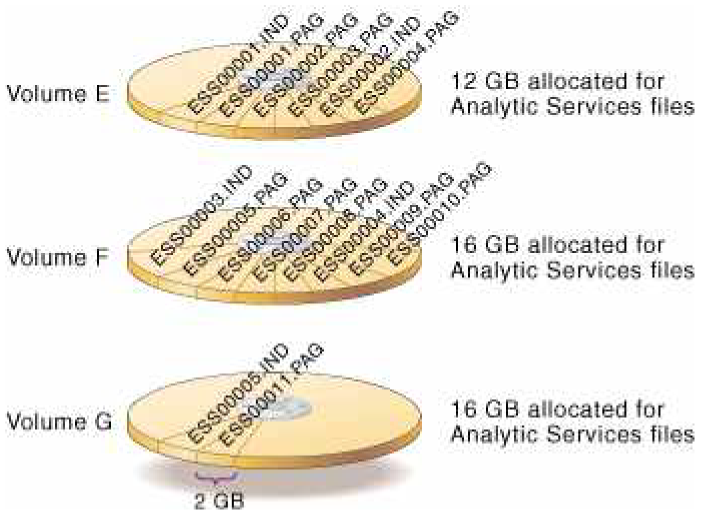
For example, suppose you want to use up to 12 GB for Essbase files on volume E, 16 GB on volume F, and 16 GB on volume G. Essbase creates a file on volume F when the sizes of the index and data files reach 12 GB on volume E and more data needs to be written out to disk.

- If the size of an individual index or data file on any volume reaches 2 GB. In the above example, suppose volumes E and F have reached their capacities and Essbase is using volume G. Figure 151 illustrates this example.

  On volume G, Essbase creates file `ess00001.ind` and fills it to the default limit of 2 GB. On volume G, Essbase creates file `ess00001.pag` and fills it to 1 GB.

  You have specified a limit of 16 GB on volume G, and you have used 3 GB. You have 13 GB left to use on volume G, but `ess00001.ind` has reached the maximum file size of 2 GB. The next time Essbase needs storage space when writing index files to disk, Essbase creates a file on volume G and names it `ess00002.ind`. Essbase then fills `ess00002.ind` to its 2 GB limit and creates `ess00003.ind`. Essbase follows the same procedures for data files.

Figure 151 Example of How Essbase Stores Files Across Volumes



Essbase names files consecutively, starting with `ess00001.`*xxx*, where *xxx* is `ind` for an index file and `pag` for a data file, and continuing up to `ess65535.`*xxx*. This naming convention applies to each volume, so in the above example, volumes E, F, and G each have files named `ess00001.pag` and `ess00001.ind`.

Keep in mind the following guidelines when specifying disk volumes:

- Specify the disk volumes in the order in which you want the volumes to be used. You need not specify the volume on which Essbase is installed as one of the volumes; you can install on one volume and store data on other volumes.
- If a volume reaches capacity, Essbase moves to the next volume.
- If all specified volumes reach capacity, Essbase stops ongoing database operations, issues an error message, and performs fatal error handling. For more information, see "Understanding Fatal Error Handling" on page 1015. If these events occur, shut down the database, allocate more disk space, and restart the database.

- You can tell Essbase to stop storing files on a volume. Essbase can still access the volume as needed, but it no longer stores additional index and data information on the volume. To stop storing information on a volume, select the volume definition that you want to remove and click Delete.
- You set disk volumes on a per-database basis. Multiple databases can use space on the same volume, so allocate space carefully. For example, if you specify 7 GB on Volume A for Database 1 and 7 GB on Volume A for Database 2, you have allocated 14 GB for Essbase files on Volume A.
- For new files, changes to the disk volumes setting take effect when you next start the database. Existing files and volumes are not affected.

Specifying Disk Volumes with Administration Services

➤ To specify disk volumes with Administration Services, see "Setting Disk Volumes" in *Oracle Essbase Administration Services Online Help*.

Specifying Disk Volumes with ESSCMD

➤ To allocate a new volume, see the ESSCMD `SETDBSTATEITEM 23` in the *Oracle Essbase Technical Reference*.

ESSCMD prompts you for the number of new disk volumes you want to add, unless you supply the number on the command line.

Then, for each new volume, ESSCMD prompts you for the following values, unless you supply them on the command line.

- Volume name (for each volume)
- Volume size (maximum space to use on the volume)—The default value is Unlimited; the minimum setting is 8 MB

When you use ESSCMD, you can specify volume size in bytes (B), kilobytes (K), megabytes (M), gigabytes (G), or terabytes (T). ESSCMD displays minimum, maximum, and current values and 0 for unlimited.

- File type—You can specify index files, data files, or both. The default is 3 - Index + Data (index and data files on the same volume).
- File size (maximum size that each file specified in file type can attain before Essbase creates a file)—The default value is 2 GB; the minimum setting is 8 MB.

The following example allocates up to 10 GB on Volume E, sets a maximum file size of 2 GB, and specifies that data files should be stored only on E:

```
SETDBSTATEITEM 23 "SAMPLE" "BASIC" "1" "E" "10G" "2" "2G"
```

➤ To change the settings on an allocated volume, enter SETDBSTATEITEM 24 in ESSCMD and either follow the prompts or supply the required values on the command line.

ESSCMD prompts you for the following values, unless you supply them on the command line:

- Volume number. (Use the GETDBSTATE command in ESSCMD to see a list of the currently defined disk volumes and to see the number assigned to each volume.)
- Volume name
- Volume size
- File type
- File size

The following example allocates up to 20 GB on Volume C and sets a maximum file size of 2 GB:

```
SETDBSTATEITEM 24 "SAMPLE" "BASIC" "1" "C" "20G" "3" "2G"
```

➤ To stop Essbase from storing additional files on a volume, enter SETDBSTATEITEM 25 in ESSCMD and either follow the prompts or supply the required values on the command line. Essbase continues accessing files on the deallocated volume but does not write new files to it.

ESSCMD prompts you for the following value, unless you supply it on the command line—Delete which volume definition. Use the GETDBSTATE command in ESSCMD to see a list of the currently defined disk volumes and to see the number assigned to each volume.

The following example deallocates the volume that is specified as fourth:

```
SETDBSTATEITEM 25 "SAMPLE" "BASIC" "4"
```

Note:

If you delete an application or database, Essbase does not remove the directory containing the application or database on a disk volume. The computer's operating system still shows the folder and file labels on the disk. However, you can reuse the same name of the application or database that you had removed on the disk volume.

For more syntax information, see the *Oracle Essbase Technical Reference*.

On UNIX, *`volume_name`* is a mounted UNIX file system. You must enter a fully qualified pathname to the Essbase installation directory (*`ESSBASEPATH`*). Essbase automatically appends the `app` directory to the path; you do not specify the `app` directory.

Consider the following example:

```
/vol2/EssbaseServer 10M
```

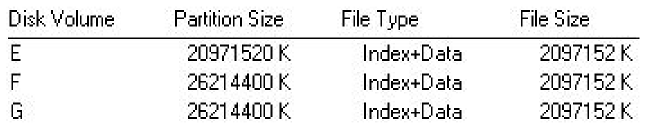
Volume size is the maximum space, in KB, allocated to the volume. The default value is unlimited—Essbase uses all available space on that volume.

Reviewing an Example of Specifying Volumes to Control Storage

Assume you want to use up to 20 GB for Essbase files on Volume E, 25 GB on Volume F, and 25 GB on Volume G. You are using the default file size limit of 2 GB.

When you load data, Essbase stores up to 20 GB on Volume E; if the database is larger than 20 GB, Essbase stores the next 25 GB on Volume F, and so on.

Figure 152 Example of Using Disk Volumes



Disk Volume	Partition Size	File Type	File Size
E	20971520 K	Index+Data	2097152 K
F	26214400 K	Index+Data	2097152 K
G	26214400 K	Index+Data	2097152 K

Data Compression

Essbase allows you to choose whether data blocks that are stored on disk are compressed, as well as which compression scheme to use. When data compression is enabled, Essbase compresses data blocks when it writes them out to disk. Essbase fully expands the compressed data blocks, including empty cells, when the blocks are swapped into the data cache.

Generally, data compression optimizes storage use. You can check compression efficiency by checking the compression ratio statistic. See "Checking the Compression Ratio" on page 777 for a review of methods.

Essbase provides several options for data compression:

- Bitmap compression, the default. Essbase stores only nonmissing values and uses a bitmapping scheme.
- Run-length encoding (RLE). Essbase compresses repetitive, consecutive values, including zeros and #MISSING values.
- zlib compression. Essbase builds a data dictionary based on the actual data being compressed.
- Index Value Pair compression. Essbase applies this compression if the block density is less than 3%.
- No compression. Essbase does not compress data blocks when they are written to disk.

Because Essbase compresses data blocks as they are written to disk, it is possible for bitmap, RLE, and uncompressed data blocks to coexist in the same data file. Keep in mind the following rules:

- When a compressed data block is brought into the data cache, Essbase expands the block to its full size, regardless of the scheme that was used to compress it.
- When Essbase stores a block on disk, Essbase treats the block the same whether it was compressed or uncompressed when it was brought into the data cache. In either case, Essbase compresses the block according to the specified compression type (including not compressing it if no compression is specified).

- If compression is not enabled, Essbase writes out the fully expanded block to disk.

You may want to disable data compression if blocks have very high density (90% or greater) and have few consecutive, repeating data values. Under these conditions, enabling compression consumes resources unnecessarily.

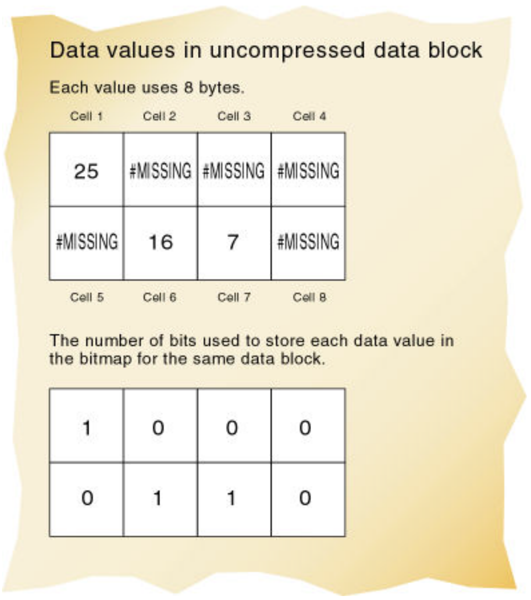
Bitmap Data Compression

With bitmap compression, Essbase uses a bitmap to represent data cells and stores only the bitmap, the block header, and the other control information. A bitmap uses one bit for each cell in the data block, whether the cell value is missing or nonmissing. When a data block is not compressed, Essbase uses 8 bytes to store every nonmissing cell.

When using bitmap compression, Essbase stores only nonmissing values and does not compress repetitive values or zeros (contrast with RLE compression, described in "RLE Data Compression" on page 774). When Essbase places a data block into the data cache, it fully expands the data block, using the bitmap to recreate the missing values.

Because the bitmap uses one bit for each cell in the data block, the bitmap scheme provides a fixed overhead for data compression. Figure 153 represents a portion of a data block as an example. In this example, Essbase uses 64 bytes to store the data in the fully expanded block but uses 1 byte (8 bits) to store the bitmap of the compressed data on disk. (Essbase also uses a 72-byte block header for each block, whether or not the block is compressed.)

Figure 153 Bitmap Data Compression


Data values in uncompressed data block
Each value uses 8 bytes.
Cell 1
Cell 2
Cell 3
Cell 4
25
#MISSING
#MISSING
#MISSING
#MISSING
16
7
#MISSING
Cell 5
Cell 6
Cell 7
Cell 8
The number of bits used to store each data value in the bitmap for the same data block.
1
0
0
0
0
1
1
0

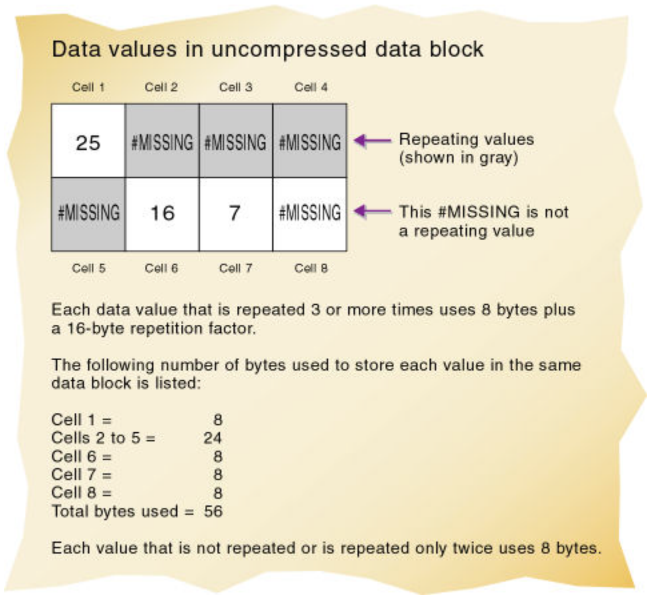
In most cases, bitmap compression conserves disk space more efficiently. However, much depends on the configuration of the data.

RLE Data Compression

When using the run-length encoding (RLE) compression scheme, Essbase compresses any consecutive, repetitive values—any value, including zero, that repeats three or more times consecutively. Essbase tracks each repeating value and the number of times it repeats consecutively.

In the example in Figure 154, Essbase uses 64 bytes to store the data in the fully expanded block but uses 56 bytes to store the compressed data on disk. (Essbase also uses a 72-byte block header for each block, whether or not the block is compressed.)

Figure 154 RLE Data Compression


Data values in uncompressed data block
Cell 1
Cell 2
Cell 3
Cell 4
25
#MISSING
#MISSING
#MISSING
Repeating values (shown in gray)
#MISSING
16
7
#MISSING
This #MISSING is not a repeating value
Cell 5
Cell 6
Cell 7
Cell 8

Each data value that is repeated 3 or more times uses 8 bytes plus a 16-byte repetition factor.

The following number of bytes used to store each value in the same data block is listed:

Cell 1 =	8
Cells 2 to 5 =	24
Cell 6 =	8
Cell 7 =	8
Cell 8 =	8
Total bytes used =	56

Each value that is not repeated or is repeated only twice uses 8 bytes.

zlib Compression

This method is used in packages such as PNG, Zip, and gzip. Calculation and data loading are faster with direct I/O and zlib compression than with buffered I/O and zlib compression. If data storage is your greatest limiting factor, use zlib, but be aware that, under some circumstances, data loads may be up to 10% slower than bitmap compression. The size of the database, however, is generally significantly smaller when you use zlib compression.

In contrast to bitmap compression, which uses an algorithm to track which values are missing and does not interact with any other type of data, zlib compression builds a data dictionary based on the actual data being compressed (including any missing values). Therefore, zlib compression should provide greater compression ratios over bitmap compression, given extremely dense data. However, because the effectiveness of the zlib algorithm is dependent (at the bit level) on the actual data being compressed, general guidelines about when zlib compression provides greater compression than bitmap compression based solely on density are not available. Unlike other compression methods, the storage space saved has little or no relationship to the number of missing cells or the number of contiguous cells of equal value. Generally, the more dense or heterogeneous the data is, the better zlib will compress it in comparison to bitmap or RLE compression. However, under some circumstances, it is possible that zlib will not yield better results than using bitmap or RLE compression. It is best to test with a representative data sample.

To estimate the storage savings you may obtain with zlib, create a small database using your usual compression technique (bitmap or RLE) with a small sampling of real data and shut down Essbase Server. Note the size of the created data files. Then clear the data in the sample database, change the compression setting to zlib, reload the same sample data, and shut down Essbase Server again. Now note the difference in the storage used. You can also use the small sample database to estimate any changes in calculation or data loading speed.

Index Value Pair Compression

Index Value Pair addresses compression on databases with larger block sizes, where the blocks are highly sparse. This compression algorithm is not selectable but is automatically used whenever appropriate by the database. The user must still choose between the compression types None, bitmap, RLE, and zlib through Administration Services.

For example, if the user selects RLE, Essbase reviews each block and evaluates the following compression types for highest compression: RLE, bitmap, or Index Value Pair. If the user chooses zlib, for example, zlib is the only compression type applied.

The following table illustrates the available compression types the user can choose and the compression types that Essbase evaluates and then applies.

Chosen Compression Type	Evaluated Compression Type
None	None
Bitmap	Bitmap, Index Value Pair
RLE	RLE, Bitmap, Index Value Pair
zlib	zlib

Deciding Which Compression Type to Use

You can choose from four compression settings: bitmap (the default), RLE, zlib, or None.

In most cases, you need not worry about choosing a setting. Bitmap compression almost always provides the best combination of fast performance and small data files. However, much depends on the configuration of the data.

Data compression is CPU-intensive. Consider the tradeoffs of computation costs versus I/O costs and disk space costs when choosing a compression setting.

In general, a database compresses better using the RLE setting than the bitmap setting if a large number of repeated nonmissing data cells for a given block have the same value. Using RLE compression is computationally more expensive than using bitmap compression. If your database shrinks significantly using RLE compression, however, you may see a performance improvement due to decreased I/O costs.

Databases usually shrink when using zlib compression, but not always. Using zlib compression significantly increases CPU processing. For most databases, this extra processing outweighs the benefits of the decreased block size. But if your database shrinks significantly using zlib compression, you may see a performance improvement due to decreased I/O costs.

The None compression setting does not reduce the disk usage of a database compared to bitmap compression. In fact, no compression may make no difference to improve the performance of the database, because bitmap compression is extremely fast.

Remember that each database is unique, and the previous statements are general characteristics of compression types. Although the default bitmap compression works well for most databases, the best way to determine the best compression setting for your database is to try each one.

Changing Data Compression Settings

Changes to the data compression setting take effect immediately as Essbase writes data blocks to disk. For blocks already on disk, Essbase does not change compression schemes or enable or disable compression. When you change the data compression settings of blocks already on disk, Essbase uses the new compression scheme the next time Essbase accesses, updates, and stores the blocks.

➤ To view or change the current settings, use a tool:

Tool	**Topic**	**Location**
Administration Services	Selecting a Data Compression Method	*Oracle Essbase Administration Services Online Help*
MaxL	**alter database**	*Oracle Essbase Technical Reference*
ESSCMD	To enable or disable data compression: SETDBSTATE or: SETDBSTATEITEM 14 To set the data compression type: SETDBSTATEITEM 15	*Oracle Essbase Technical Reference*

Example of Using SETDBSTATEITEM

➤ To enable or disable data compression, enter SETDBSTATEITEM 14 in ESSCMD and either follow the prompts or supply the required values on the command line.

ESSCMD prompts you for the following values, unless you supply them on the command line:

- Data Compression on Disk? Enter Y (Yes, the default) or N (No).
- Data Compression Type. Enter 1 (run-length encoding) or 2 (bitmap, the default).

➤ To specify the data compression type, enter SETDBSTATEITEM 15 in ESSCMD and either follow the prompts or supply the required values on the command line. ESSCMD prompts you for a value of “1” (run length encoding) or “2” (bitmap, the default).

The following example enables Bitmap compression:

```
SETDBSTATEITEM 14 "SAMPLE" "BASIC" "Y" "2"
```

For more syntax information, see the *Oracle Essbase Technical Reference*.

Checking the Compression Ratio

The compression ratio represents the ratio of the compressed block size (including overhead) to the uncompressed block size, regardless of the compression type in effect. Overhead is the space required by mechanisms that manage compression/expansion.

➤ To check the compression ratio, use a tool:

Tool	Topic	Location
Administration Services	Checking the Compression Ratio	*Oracle Essbase Administration Services Online Help*
ESSCMD	GETDBSTATS	*Oracle Essbase Technical Reference*

Note:

The larger the number, the more compression. The compression ratio can vary widely from block to block.

Data Block Size

Data block size is determined by the amount of data in a particular combination of dense dimensions. For example, when you change the dense or sparse configuration of one or more dimensions in the database, the data block size changes. Data block size is 8*n* bytes, where *n* is the number of cells that exist for that combination of dense dimensions.

Note:

The optimum size range is 8 KB–100 KB.

For information about determining the size of a data block, see “Size of Expanded Data Block” on page 1023.

➤ To view the block size for a database, use a tool:

Tool	**Topic**	**Location**
Administration Services	Checking Data Block Statistics	*Oracle Essbase Administration Services Online Help*
ESSCMD	GETDBSTATS	*Oracle Essbase Technical Reference*

48 Ensuring Data Integrity

In This Chapter

The information in this chapter applies only to block storage databases and is not relevant to aggregate storage databases. Also see Chapter 57, “Comparison of Aggregate and Block Storage.”

Understanding Transactions

When a database is in read/write mode, Essbase considers every update request to the server (such as a data load, a calculation, or a statement in a calculation script) as a transaction. Essbase tracks information about transactions in a transaction control file (`dbname.tct`).

The transaction control file contains an entry for each transaction and tracks the current state of each transaction (Active, Committed, or Aborted).

See “Understanding How Essbase Handles Transactions” on page 786.

Understanding Isolation Levels

Isolation levels determine how Essbase commits data to disk. When data is committed, it is taken from server memory and written to the database on disk. Essbase automatically commits data to disk. There are no explicit commands that users perform to commit data blocks. However, setting the isolation level for a database defines how Essbase automatically commits data blocks.

Essbase offers two isolation levels for transactions—*committed access* and *uncommitted access* (the default). You can optimize data integrity by using committed access.

For an explanation of access types, see "Committed Access" on page 781 and "Uncommitted Access" on page 783.

Note:

Setting the isolation level to committed access may increase memory and time requirements for database restructure.

Note:

The Spreadsheet Add-in lock and Send and the Grid API are always in Committed Access Mode.

Data Locks

Essbase issues write (exclusive) locks for blocks that are created, updated, or deleted, and issues read (shared) locks for blocks that should be accessed but not modified. By issuing the appropriate locks, Essbase ensures that data changed by one operation cannot be corrupted by a concurrent update.

This section discusses locks on data blocks, not locks on database artifacts. For information about locking and unlocking outlines and other artifacts, see "Locking and Unlocking Artifacts" on page 719.

Table 89 explains the lock types:

Table 89 Basic Lock Types

Lock	Description
Write (exclusive) lock	Prevents any other transaction from accessing the locked data block. Used for all data block updates, including spreadsheet lock and send operations.
Read (shared) lock	Allows other transactions read-only access to the data block but prevents other transactions from modifying the data block.

Table 90 shows the locks that Essbase issues for various types of operations.

Table 90 Locking by Higher-Level Functions

Type of Operation	Lock Issued
Spreadsheet retrieve	Read (shared) lock on each data block.
Retrieve and lock	Write (exclusive) lock on all affected blocks. A subsequent send command commits the data.
Calculate derived block	Write lock on the block being calculated. As a block is calculated, all blocks containing the block's children acquire read locks.
Data load	Write lock
Restructure	Write lock

How Essbase handles locking depends on whether committed or uncommitted access is enabled.

Committed Access

Committed access provides a high level of data consistency because only one transaction at a time can update data blocks. Under committed access, Essbase allows transactions to hold read/write locks on all data blocks involved with the transaction until the transaction completes and commits. However, you can still allow read-only access to the last committed data values.

Essbase provides options that determine when locks are issued on data blocks:

- Pre-image access (enabled by default). Pre-image access provides users read-only access to data blocks that are locked for the duration of a concurrent transaction. Users see the last committed data values for the locked data blocks.
- Wait, or timeout:
  - Indefinite wait (the default). The transaction waits to acquire a lock on the required locked block.
  - Immediate access, or no wait. If a required block is locked by another transaction, Essbase displays a lock timeout message, and the transaction aborts.
  - A number of seconds that you specify. The transaction waits that number of seconds to acquire a lock on the required locked blocks. If the specified time runs out before the transaction acquires a lock, Essbase displays a lock timeout message, and the transaction aborts.

When pre-image access is enabled, you are not limited to read-only access to data blocks; if you need write access to locked blocks, the transaction waits for write access or times out, depending on the wait or timeout setting. The transaction gets immediate write access to data blocks that are not locked by another transaction.

If pre-image access is not enabled, and if you need read or write access to locked blocks, the transaction waits for write access or times out, depending on the wait or timeout setting.

Memory Considerations with Committed Access

Under committed access, note the following memory considerations:

- Essbase retains redundant data until a transaction commits. Allow disk space for double the size of the database to accommodate redundant data.
- Models with a large number of blocks may experience memory problems under committed access. Each lock (one lock per block) uses approximately 80 bytes of memory per calculation, and each lock is held in memory until the transaction is complete. There is a limit to the addressable memory space per process, and eventually models with a large number of blocks may hit this limit, causing the transaction to terminate. In such cases, consider using Uncommitted Access.

Note:

Setting the isolation level to committed access may increase memory and time requirements during database restructure.

Locking Under Committed Access

Under committed access, Essbase locks blocks for read and write access:

- For read access, the lock remains until another transaction requests it, whether or not the transaction is complete. Other transactions can read the locked block, but none can alter it.
- For write access, a transaction locks and holds the lock on each block that it modifies until the transaction completes.

Table 91 illustrates locking behavior under committed access when multiple transactions are contending for a lock on the same data. In the example in Table 91, transaction Tx1 is running, and transaction Tx2 is requesting access to the same data.

Note that access to locked blocks depends on what options are enabled. For a discussion of options, see "Committed Access" on page 781.

Table 91 Locking Behavior Under Committed Access

		Tx1 holds read lock; **Tx2 requests read lock**	**Tx1 holds read lock;** **Tx2 requests write lock**	**Tx1 holds write lock;** **Tx2 requests read lock**	**Tx1 holds write lock;** **Tx2 requests write lock**
Pre-image access enabled	Wait (timeout) period specified (indefinite wait or a number of seconds wait)	Tx2 gets read lock.	Tx2 waits for Tx1 to release read lock.	Tx2 gets pre-image access.	Tx2 waits for Tx1 to release write lock.
	No wait (timeout) period specified (immediate timeout)	Tx2 gets read lock.	Essbase issues timeout message and aborts the transaction.	Tx2 gets pre-image access.	Essbase issues timeout message and aborts the Tx2 transaction.
No pre-image access	Wait (timeout) period specified (indefinite wait or a number of seconds wait)	Tx2 gets read lock.	Tx2 waits for Tx1 to release read lock.	Tx2 waits for Tx1 to release write lock.	Tx2 waits for Tx1 to release write lock.
	No wait (timeout) period specified (immediate timeout)	Tx2 gets read lock.	Essbase issues time-out message and aborts the Tx2 transaction.	Essbase issues timeout message and aborts the Tx2 transaction.	Essbase issues timeout message and aborts the Tx2 transaction.

For information about how to set concurrency parameters, see "Specifying Data Integrity Settings" on page 787.

Concurrency with Committed Access

Occasionally, under committed access, a deadlock results when two transactions are locking or waiting for access to the same blocks, and neither transaction can complete under these conditions.

For example, if transaction Tx1 needs to update first data block B1 and then data block B2, it first locks B1 and then attempts to lock B2. Meanwhile, if transaction Tx2 needs to update first data block B2 and then block B1, Tx2 first locks B2 and then attempts to lock B1. Tx1 locked B1 and is waiting for B2, and Tx2 locked B2 and is waiting for B1.

Essbase transactions periodically perform deadlock detection while waiting to acquire a lock. If detected, Essbase issues an error message, and the transaction fails.

If you try to update a block that is locked by another user, Essbase behaves in these ways:

- If wait is set to indefinite, the transaction waits to acquire the needed locks.
- If wait is set to 0 (immediate), and if the required blocks are not immediately available, Essbase displays an error message, and the transaction fails.
- If wait is set to a user-specified number of seconds, and the time has expired, Essbase displays an error message and aborts the transaction.
- If the request times out, try the operation again.

For information about how to set concurrency options, see “Specifying Data Integrity Settings” on page 787.

Rollback with Committed Access

Under committed access, if the server crashes, Essbase rolls back all database updates by transactions that were in progress when the server stopped, ensuring that changes made by the aborted transactions are undone.

If a transaction is aborted because of a nonfatal error, all changes made by the transaction are rolled back.

See “Recovering from a Crashed Database” on page 789.

Uncommitted Access

With uncommitted access (enabled by default), the Essbase kernel allows transactions to hold read/write locks on a block-by-block basis; Essbase releases a block after it is updated but does not commit blocks until the transaction completes or until a specified limit (a “synchronization point”) has been reached. You can set this limit, as described below.

Concurrent users accessing the same data blocks might experience unexpected results under uncommitted access, because Essbase allows read-only access to data at its last commit point.

With uncommitted access, you can control when Essbase performs an explicit commit operation by specifying synchronization point parameters:

- Commit Blocks (number of blocks modified before a synchronization point occurs). The default is 3,000.

  If you set Commit Blocks to 0, the synchronization point occurs at the end of the transaction.
- Commit Rows (number of rows to data load before a synchronization point occurs). The default is 0, which means that the synchronization point occurs at the end of the data load.
- If either Commit Blocks or Commit Rows has a nonzero value, a synchronization point occurs when the first threshold is reached. For example, if Commit Blocks is 10 but Commit Rows is 0 and you load data, a synchronization point occurs after 10 blocks are updated. If Commit Blocks is 5 and Commit Rows is 5 and you load data, a synchronization point occurs after 5 rows are loaded or 5 blocks are updated, whichever happens first.

If a user-defined threshold is exceeded during an operation, Essbase issues a synchronization point to commit the data processed to that point. Essbase performs as many synchronization points as are necessary to complete the operation.

Note:

Essbase analyzes the value of Commit Blocks and Commit Rows during its analysis of feasibility for parallel calculation use. If Essbase finds the values set too low, it automatically increases them.

For information about how to specify synchronization point parameters, see "Specifying Data Integrity Settings" on page 787.

Caution!

Essbase retains redundant data to enforce transactional semantics. Allow disk space for double the size of the database to accommodate redundant data, particularly if both Commit Blocks and Commit Rows are set to 0.

Memory Considerations with Uncommitted Access

If your data cache is too small to hold the number of blocks specified in your Commit Blocks and Commit Rows settings, the blocks will be written to disk as soon as the caches become full, which will be before the transaction is committed.

Locking Under Uncommitted Access

Under uncommitted access, Essbase locks blocks for write access until Essbase finishes updating the block. Under committed access, Essbase holds locks until a transaction completes.

Table 92 illustrates locking behavior under uncommitted access when many transactions contend for a lock on the same data. In the example in Table 92, transaction Tx1 is running and transaction Tx2 is requesting access to the same data.

Table 92 Locking Behavior with Uncommitted Access

Status when Tx2 Makes a Request	If Tx1 holds read lock	If Tx1 holds write lock
Read lock	Tx2 gets read lock	Tx2 gets read lock
Write lock	Tx2 gets write lock	Tx2 waits for Tx1 to release the lock

Concurrency with Uncommitted Access

With uncommitted access, blocks are released more frequently than with committed access, when all blocks are locked until the end of the transaction.

Rollback with Uncommitted Access

Under uncommitted access, if the server crashes, Essbase rolls back all database updates from the point of the last successful commit. Some updates from an aborted transaction may have committed. Whether transactions committed their updates the way users expected depends on the order in which overlapping transactions updated and committed data.

If a transaction is aborted because of a nonfatal error, Essbase commits only the data that the transaction finished processing before the abort of the transaction.

See "Recovering from a Crashed Database" on page 789.

Parallel Calculation and Uncommitted Access

If Essbase is using parallel calculation, Essbase checks the commit threshold.

Committed Versus Uncommitted Access

Consider the following issues when choosing an isolation level:

Table 93 Issues Affecting Selection Of An Isolation Level

Issue	Explanation
Database performance	Uncommitted access always yields better database performance than committed access. When using uncommitted access, Essbase does not create locks that are held for the duration of a transaction but commits data based on short-term write locks.
Data consistency	Committed access provides a higher level of data consistency than uncommitted access. Retrievals from a database are more consistent. Also, only one transaction at a time can update data blocks when the isolation level is set to committed access. This factor is important in databases where multiple transactions attempt to update the database simultaneously.
Data concurrency	Uncommitted access provides better data concurrency than committed access. Blocks are released more frequently than during committed access. With committed access, deadlocks can occur.
Database rollbacks	If a server crash or other server interruption occurs during active transactions, the Essbase kernel rolls back the transactions when the server is restarted. With committed access, rollbacks return the database to its state before

Issue	Explanation
	transactions began. With uncommitted access, rollbacks may result in some data being committed and some data not being committed. See "What to Expect if a Server Interruption Occurs" on page 791.

Understanding How Essbase Handles Transactions

Essbase tracks transactions from start to finish, swapping data blocks in and out of memory as needed and committing data blocks when a transaction completes. The following list describes how Essbase handles a transaction: all list items apply to committed and uncommitted access (see "Understanding Isolation Levels" on page 779).

1. A user or batch program begins an operation.
2. The OLAP engine notifies the Essbase kernel that a transaction is to begin.
3. The Essbase kernel begins the transaction.
4. The OLAP engine requests data from the Essbase kernel.
5. The Essbase kernel locates the requested data. It passes the data, and some associated control information, to the OLAP engine. If you are using Spreadsheet Add-in, this data is displayed on the sheet.
6. If you are using Spreadsheet Add-in, when you modify data, you issue the Send command.
7. The Essbase kernel associates the transaction with an entry in its transaction control table.
8. After the operation is complete on the OLAP engine side, the OLAP engine notifies the Essbase kernel about the update, and the Essbase kernel updates internal data structures accordingly.
9. Steps 4–8 repeat as often as necessary to complete the operation.
10. The transaction ends. If Essbase encounters an error during transaction processing, it aborts the transaction. If no errors are encountered, Essbase commits the transaction. For differences in commit behavior under committed and uncommitted access, see "Understanding Isolation Levels" on page 779.
11. Essbase issues a message to notify the client that the transaction is complete; for example, "TOTAL CALC ELAPSED TIME..."

Under uncommitted access, it is possible to access uncommitted data when multiple transactions are active and are accessing the same data. Transaction results are unpredictable under uncommitted access.

Under uncommitted access, if you have defined a commit threshold, Essbase may need to break down one database operation into multiple synchronization points. See "Uncommitted Access" on page 783 for information on commit thresholds.

Specifying Data Integrity Settings

You can specify isolation level, synchronization point parameters, and concurrency parameters using Administration Services, MaxL, or ESSCMD. Changes to isolation level settings take effect the next time there are no active transactions. For information about deciding which settings to choose, see "Committed Access" on page 781 and "Uncommitted Access" on page 783.

➤ To specify data integrity settings, use a tool:

Tool	Topic	Location
Administration Services	Setting Data Integrity Options	*Oracle Essbase Administration Services Online Help*
MaxL	**alter database**	*Oracle Essbase Technical Reference*
ESSCMD	SETDBSTATEITEM 18	*Oracle Essbase Technical Reference*

Example of Specifying Isolation Level Settings with ESSCMD

➤ To specify isolation level settings using ESSCMD, enter SETDBSTATEITEM 18 in ESSCMD and either follow the prompts or supply the required values on the command line.

Choose 1 (committed access) or 2 (uncommitted access, the default). Depending on which you specify, ESSCMD prompts you for other parameters (or you can supply the values on the command line).

If you choose 1 (committed access), ESSCMD prompts for the following information:

- Pre-image access; Y (Yes) or N (No, the default). Pre-image access provides users read-only access to data blocks that are locked for the duration of a transaction. Users see the last committed data values for the locked data blocks.
- Wait (in the Database Settings dialog box) or timeout (in ESSCMD): -1, 0, or *n*.
  - -1 is indefinite wait.
  - 0 is immediate access, or no wait.
  - *n* is the number of seconds that you specify.

If you choose 2 (uncommitted access), ESSCMD prompts for the following values. See "Uncommitted Access" on page 783 for explanations of these options.

- Number of blocks modified before internal commit
- Number of rows to data load before internal commit

You can also specify isolation level parameters (pre-image access and so on) by specifying parameters 19–22 on SETDBSTATEITEM. Enter SETDBSTATEITEM with no parameters; ESSCMD displays a list that includes each parameter by number, with a description.

Here is an example of using SETDBSTATEITEM to set an isolation level. This example enables committed access and pre-image access and specifies indefinite wait time.

```
SETDBSTATEITEM 18 "SAMPLE" "BASIC" "1" "Y" "-1"
```

See the *Oracle Essbase Technical Reference*.

Example of Specifying Isolation Level Settings with MaxL

To specify isolation level settings, use this MaxL statement:

```
alter database appname.dbname enable committed_mode
```

See the *Oracle Essbase Technical Reference*.

Accommodating Data Redundancy

To ensure data integrity, the Essbase kernel temporarily retains redundant (duplicate) information. To accommodate redundant information, allow disk space for double the size of the database.

Essbase maintains a file called *dbname*.esm, in which it stores crucial control information.

Caution!

The *dbname*.tct file, *dbname*.esm file, the index files, and the data files contain information crucial for data recovery. Never alter or delete these files.

Checking Structural and Data Integrity

To validate database integrity and to check for database corruption, perform an action:

- Perform a dense restructure. Because a dense restructure recreates all blocks within a database, this method verifies index nodes and cells for each block.
- Export all levels of data from the database. Exporting an entire database accesses blocks and all data values across the entire database.
- Use the ESSCMD VALIDATE command to check structural and data integrity. See "Using VALIDATE to Check Integrity" on page 788.

If errors occur during any of these checks, restore the database from backups. See the *Oracle Hyperion Enterprise Performance Management System Backup and Recovery Guide*.

Using VALIDATE to Check Integrity

The VALIDATE command performs many structural and data integrity checks:

- Verifies the structural integrity of free space information in the index.
- Compares the data block key in the index page with the data block key in the corresponding data block.

- The Essbase index contains an entry for every data block. For every read operation, VALIDATE automatically compares the index key in the index page with the index key in the corresponding data block and checks other header information in the block. If it encounters a mismatch, VALIDATE displays an error message and continues processing until it checks the entire database.
- Restructures data blocks whose restructure was deferred with incremental restructuring.
- Checks every block in the database to make sure each value is a valid floating point number.
- Verifies the structural integrity of the LROs catalog.

Note:

When you issue the VALIDATE command, we recommend placing the database in read-only mode.

As Essbase encounters mismatches, it records error messages in the VALIDATE error log. You can specify a file name for error logging; Essbase prompts you for this information if you do not provide it. The VALIDATE utility runs until it has checked the entire database.

You can use the VALIDATE command in ESSCMD to perform these structural integrity checks.

During index free space validation, the VALIDATE command verifies the structural integrity of free space information in the index. If integrity errors exist, Essbase records them in the VALIDATE log. The file that you specified on the VALIDATE command holds the error log.

If VALIDATE detects integrity errors regarding the index free space information, the database must be rebuilt. You can rebuild in three ways:

- Restore the database from a recent system backup
- Restore the data by exporting data from the database; creating an empty database; and loading the exported data into the new database.
- Restructure the database

See Chapter 53, "Optimizing Database Restructuring" and the *Oracle Hyperion Enterprise Performance Management System Backup and Recovery Guide*.

Even if you do not use VALIDATE, Essbase automatically performs certain validity checking whenever a read operation is performed, to ensure that the index is synchronized with the data.

For every read operation, Essbase compares the data block key in the index page with the data block key in the corresponding data block and checks other header information in the block.

If Essbase encounters a mismatch, it displays an "Invalid block header" error message.

Recovering from a Crashed Database

After a server interruption such as a crash, Essbase recovers a database, rolling back all transactions that were active when the interruption occurred. Recovery time depends on the index size. The larger the index, the longer it takes.

Essbase also recovers and consolidates free fragments (unused addressable units in the data blocks). However, free space recovery is the most time-consuming aspect of database recovery, so it is delayed by default. You must trigger free space recovery explicitly unless you have changed the default setting. See "Free Space Recovery" on page 790 for the advantages and disadvantages of delaying free space recovery.

Essbase recovers data as soon as the server is started after a server interruption. Recovery phases:

1. Transaction recovery rolls back all transactions that were active when the interruption occurred.
2. Index file recovery truncates files to their previously committed sizes.
3. Data free space recovery rebuilds the data free space tables. The size of the index determines the duration of this phase.

Note:

Free space recovery is delayed until you trigger it, unless you have changed the default setting.

A media failure (faulty disk, disk failure, or head crash) requires that you restore data from backups. See the *Oracle Hyperion Enterprise Performance Management System Backup and Recovery Guide*.

Caution!

Do not move, copy, modify, or delete the following files: `ess`*`xxxxx`*`.ind`, `ess`*`xxxxx`*`.pag`, *`dbname`*`.ind`, *`dbname`*`.esm`, *`dbname`*`.tct`. Doing so can result in data corruption.

The Essbase kernel uses fatal error handling to display appropriate messages and to shut down the server, depending on the error encountered. For an explanation of how fatal error handling works, see "Understanding Fatal Error Handling" on page 1015.

For information about how transactions are rolled back after a crash, see "Committed Versus Uncommitted Access" on page 785.

Free Space Recovery

Database recovery occurs when you load an application that has just crashed or terminated abnormally. Essbase does not perform free space recovery automatically, because it is the most expensive part of database recovery. You must either trigger free space recovery explicitly or change the default setting so that Essbase will recover free space automatically.

All database functions run normally whether you recover free space or not. When you recover free space, you can reuse disk space marked as free in the data files. Keep in mind that free space recovery is time-consuming, so you might delay it until a better time.

You should, however, perform free space recovery as soon as possible to take advantage of the free space in the data files and to ensure that the database has not been corrupted. Also, if a database crashes repeatedly, and you do not run free space recovery, data files can become unnecessarily large.

To trigger free space recovery, use the MaxL alter database command. For example:

```
alter database DBS-NAME recover freespace
```

See the *Technical Reference*.

To change the default behavior for free space recovery, change the DELAYEDRECOVERY configuration setting to FALSE. See the "Configuration Settings" section of the *Technical Reference*.

To get information about free space recovery, use the GETDBSTATS command. GETDBSTATS provides the following information about free space recovery:

```
Free Space is Recoverable                  : true/false
Estimated Bytes of Recoverable Free Space : nnn
```

Note:

If free space is recoverable, the block counters are estimates and do not necessarily match the number of existing blocks.

What to Expect if a Server Interruption Occurs

Table 94 lists types of Essbase Server interruptions and their results:

Table 94 Essbase Recovery Handling

Interruption	Result
● Power loss on server ● Operating system crash ● Server stopped with Ctrl+C keys	Essbase Server stops. When you restart the server, Essbase recovers the database.
● Operation cannot complete because of system limitations ● Memory shortage ● Out of disk space ● Application stopped with Ctrl+C keys	Essbase performs fatal error handling. You may need to allocate more memory or disk space and restart the server.
Server crash	Essbase Exception Manager creates an error log of type `.xcp`. Server stops. When you restart the server, Essbase recovers the database.

Table 95 shows what you must do if a server interruption occurs during a transaction. How Essbase recovers from an interruption depends on the transaction isolation level setting (committed or uncommitted access). See "Rollback with Committed Access" on page 783 and "Rollback with Uncommitted Access" on page 785.

Table 95 Recovery Procedures for Server Requests

Type of Request	Recommended Action
Lock (for spreadsheet update)	Issue the lock command again.
Send (spreadsheet update)	If Essbase issues an error, repeat the last send operation. If the spreadsheet has been lost or does not exist, and if you are using SSAUDIT spreadsheet logging, reload the `dbname.atx` file. See "How to Use Spreadsheet Update Logging" on page 792.
Calculation	Check the server and application logs to see where the calculation left off. See "Viewing the Essbase Server and Application Logs" on page 742. Decide whether to start the calculation over. Repeat the last calculation.
Data load	Perform an action: ● Repeat the last data load. See Chapter 19, "Performing and Debugging Data Loads or Dimension Builds." ● Load the error log. See "Loading Dimension Build and Data Load Error Logs" on page 755.
Arithmetic data load (adding to or subtracting from values in the database)	If the database is set to committed access, reload the data. (The transaction has been rolled back.) If the database is set to uncommitted access, some of the data loaded, so if you reload all of the data, you receive incorrect results for the data values that loaded twice. Therefore, perform the following actions: ● Clear the database. ● Restore the database to its preload state. ● Rerun the load.
Restructure	The restructure is not complete. Delete the temporary restructure files: `.pan`, `.inn`, and `.otn`. Repeat the last operation that caused a restructure.

Note:

If the UPDATECALC parameter is set to FALSE, Essbase recalculates the entire database if an interruption occurs during a calculation. (The default value of the parameter is TRUE.)

How to Use Spreadsheet Update Logging

For extra protection against data loss and for spreadsheet audit information, Essbase provides spreadsheet update logging, which you enable by using the SSAUDIT or SSAUDITR parameter in the `essbase.cfg` file on the server. You can specify SSAUDIT for all databases on the server or for individual databases. See the *Oracle Essbase Technical Reference*.

Essbase handles recovery under normal situations. However, sometimes you may want to load the spreadsheet update log manually. For example, if you have restored from a recent backup and do not want to lose changes made since the backup or you experience a media failure, you can recover transactions from the update log. To do so, use the Essbase command-line facility, ESSCMD, from the server console.

The following ESSCMD command sequence loads the update log:

```
LOGIN hostnode username password
SELECT appname dbname
LOADDATA 3 filepath:appname.ATX
EXIT
```

To simplify loading the update log, prepare a batch file as described in "Using Script and Batch Files for Batch Processing" on page 1052.

When SSAUDIT or SSAUDITR is specified, Essbase logs spreadsheet update transactions chronologically. Essbase uses two files:

- *dbname*`.atx` stores spreadsheet update transactions as a unit that can be used as the input source for data loads.
- *dbname*`.alg` contains historical information for each transaction, such as user name, date, and timestamp, and the number of transaction rows from the `.atx` file.

Both files are stored on the server.

The spreadsheet update log can get quite large, even if you are using SSAUDITR, Essbase clears the log only after you back up data. If spreadsheet updates are frequent, consider periodically deleting the log manually.

When a database is started after a shutdown, if spreadsheet logging is enabled, Essbase writes the following message to the database log:

```
Starting Spreadsheet Log
volumename\app\appname\dbname\dbname.atx for database dbname
```

An example of the message is:

```
Starting Spreadsheet Log Hyperion\products\Essbase\EssbaseServer\app\app1
\sample\sample.atx for database Sample
```

To ensure successful spreadsheet update logging, stop and restart the application after either of the following:

- Any operation that causes a restructure. See Chapter 53, "Optimizing Database Restructuring."
- Running any of the following ESSCMD commands:
    - CREATEAPP
    - CREATEDB
    - COPYDB
    - RENAMEDB

Essbase ensures that if you enable spreadsheet logging, updates cannot take place without being logged. If for any reason Essbase cannot write to the update log, Essbase stops the transaction and issues an error message.

SSAUDIT and SSAUDITR are available only from the `essbase.cfg` file.

Considering Hybrid Analysis Issues

Hybrid Analysis provides a way to integrate a relational database with a multidimensional database so that lower-level members and their associated data remain in the relational database while upper-level members and their associated data reside in the Essbase database. This option presents additional issues regarding data consistency and integrity.

For information about ensuring that the data is correct in all locations, see "Managing Data Consistency in Hybrid Analysis" on page 601.

49 Using MaxL Data Definition Language

In This Chapter

Also see the MaxL DDL section of the *Oracle Essbase Technical Reference*.

The MaxL DDL Language

The MaxL data definition language is an interface for making administrative requests to Essbase using *statements* rather than a series of commands with complicated arguments.

Using MaxL, you can automate administrative operations on Essbase databases. You can write MaxL scripts with variables to make them customizable and reusable.

For Essbase to receive and parse MaxL statements, you must "pass" them to the Essbase Server using either the MaxL Shell (`essmsh`), Administration Services, or a customized Perl program that uses the MaxL Perl Module, which enables you to embed its statements in Perl programs.

Overview of Statements

All MaxL DDL scripts and interactive sessions comprise a login and a sequence of statements, each terminated by a semicolon and consisting of grammatical sequences of keywords and variables. Statements are similar to English sentences; for example,

```
create or replace user <user-name> identified by <password>;
```

MaxL statements begin with a verb, such as *create* or *alter*, which indicates the type of operation to perform. Then you specify an artifact, such as *database* or *user*, which indicates the Essbase elements you are working with. The rest of the statement provides details about the action to perform, using a grammatically correct sequence of statement parts, or tokens.

For an overall picture of grammar requirements and the verb-artifact structure of MaxL statements, see the MaxL DDL section of the *Oracle Essbase Technical Reference*.

Components of Statements

The MaxL parser recognizes and requires an ordered presentation of *tokens*, which are the components of statements. A token is a space-delimited sequence of valid characters that is recognized by MaxL as a single readable unit. Tokens can be any of the following:

- Keywords
- Names
- Strings
- Numbers

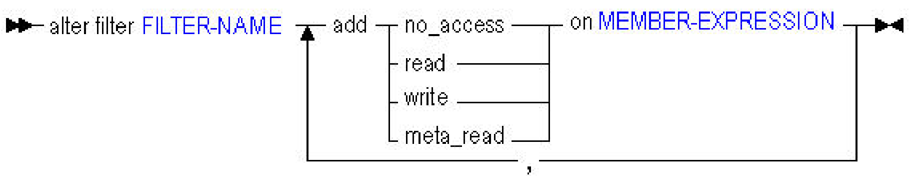
Keywords

Keywords are the reserved words that make up the MaxL vocabulary. These include verbs, artifacts, and other words needed to construct statements. Keywords in MaxL are independent of your data; conversely, you must define all other MaxL tokens (names, for example).

The MaxL parser expects to see MaxL keywords and other tokens in their correct grammatical order, as diagrammed in MaxL topics in the *Oracle Essbase Technical Reference*.

Figure 155 shows a sample syntax diagram from the *Oracle Essbase Technical Reference*. Only the lowercase words in the diagram represent keywords. The other elements are placeholders for names or values that you provide.

Figure 155 Example of MaxL Syntax Diagram: Alter Filter



Note:

Keywords are not case-sensitive; the use of lowercase for keywords is a documentation convention. For more information about how to interpret the diagrams, see “How to Read MaxL Railroad Diagrams” in the *Oracle Essbase Technical Reference*.

Names

Names in MaxL are used to uniquely identify databases and database artifacts, such as users, applications, or filters.

Rules for Names

Unless you enclose a MaxL name within single quotation marks, a MaxL name is a string that must begin with an alphabetic character or the underscore. Names that are not enclosed in quotation marks may contain only alphabetic characters, numbers, and the underscore.

When enclosed in single quotation marks, a name may contain white space and any of the following special characters:

```
.
,
;
:
%
$
"
'
*
+
-
=
<
>
[
]
{
}
(
)
?
!
 /
 \
|
 ~
`
#
&
@
^
```

Note:

Any name that is also a MaxL keyword must be enclosed in single quotation marks. For a list of keywords, see the "Reserved Words List in the MaxL DDL" section of the *Oracle Essbase Technical Reference*.

Examples:

The following application names *do not* require single quotation marks:

```
Orange
Orange22
_Orange
```

The following application names *do* require single quotation marks:

```
Orange County(because the name contains a space)
22Orange (because the name begins with a number)
variable (because the name is a MaxL keyword)
```

Types of Names

Some Essbase artifacts have single names, and some require compound names known as *doubles* and *triples*, which express the nesting of namespaces.

A *singleton name* can be meaningful in a systemwide context—the artifact to which it refers may be global to Essbase—or it needs no specified application or database context. For example, an application has a singleton name because it need not be considered in the context of another application or database.

A *double* is two names connected by a period, and a *triple* is three names connected by two periods. Doubles and triples show the inherited namespace of the named entity. For example, a database usually is identified using two names. The first identifies the application in which the database resides, and the second is the database name; for example:

```
Sample.Basic
```

Database artifacts, such as filters, usually are identified using triple names: the first two names identify the application and database, and the third is the artifact name. Therefore, a filter name could look like this:

```
sample.basic.filter3.
```

Table 96 shows the type of name required for the most common artifacts and provides an example of the name used in a statement.

Table 96 Name Requirements for Artifacts

Artifact	Name	Example
User	singleton	`create user` **`Fiona`** `identified by 'password';`
Group	singleton	`alter user Fiona add to group` **`Managers`**`;`
Host	singleton	`drop replicated partition Samppart.Company from Sampeast.East at` **`EastHost`**`;`
Application	singleton	`create application` **`'&New App'`**`;`
Database	double	`display database` **`'&New App'.testdb`**`;`
Calculation	triple	`drop calculation` **`Sample.basic.'alloc.csc'`**`;`
Filter	triple	`display filter row` **`sample.basic.filter1`**`;`
Function (local)	double	`drop function` **`sample.'@COVARIANCE'`**`;`
Function (global)	singleton	`create function` **`'@JSUM'`** `as 'CalcFnc.sum';`

Artifact	Name	Example
Location alias	triple	`drop location alias` **`Main.Sales.EasternDB`**`;`
Role	singleton	`grant` **`designer`** `on database Sample.basic to Fiona;`
Substitution variable	singleton	`alter system add variable` **`Current_month`**`;` `alter system set variable` **`Current_month`** `July;`
Disk volume	singleton to define, triple to display	`alter database AP.main1 add disk volume` **`G`**`;` `alter database AP.main1 set disk volume` **`G`** `partition_size 200mb;` `display disk volume` **`sample.basic.C`**`;`

Strings

Strings are used in MaxL statements to represent the text of comments, member expressions, calculation scripts, and file references. Strings can begin with any valid character. As with names, strings containing white space or special characters must be enclosed in single quotation marks.

See "Rules for Names" on page 796 for a list of valid special characters.

Table 97 shows examples of statement elements that are strings:

Table 97 Examples of String in MaxL Statements

Type of String	Example
Password	`create user Fiona identified by` **`sunflower`**`;`
Comment	`alter group Financial comment` **`'Reports due July 31'`**`;`
Member expression	`create filter sample.basic.filt1 read on 'Sales,`**`@ATTRIBUTE (Bottle)`**`';` `create or replace replicated partition sampeast.east area` **`'@IDESC(East), @IDESC(Qtr1)'`** `to samppart.company mapped globally '("Eastern Region")' to '(East)';`
Body of a calculation	`execute calculation` **`'"Variance"=@VAR(Actual, Budget); "Variance %"=@VARPER(Actual, Budget);'`** `on Sample.basic;`
File reference	`spool on to` **`'/homes/fiona/out.txt'`**`;`

Numbers

You use numbers in MaxL statements to change certain database settings in Essbase. For example, you can change cache and buffer sizes or set systemwide intervals such as the number of days elapsing before users are required to change their passwords. To change numeric settings, you can use positive integers, positive real numbers, and zero. Decimals and scientific notation are permitted.

Examples:

```
1000
2.2
645e-2
```

For size settings, units must follow the number. Spaces between numbers and units are optional. Units are case-insensitive and may include the following: B/b (bytes), KB/kb (kilobytes), MB/mb (megabytes), GB/gb (gigabytes), and TB/tb (terabytes). If no units are given, bytes are assumed.

Examples:

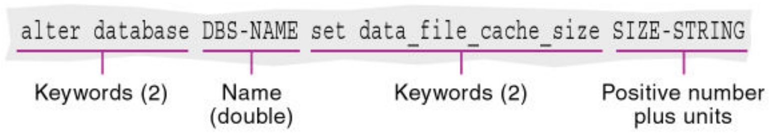
```
1000 b
5.5GB
645e-2 mb
145 KB
2,000e-3TB
```

Analysis of Statements

This section helps you review what you have learned about statements by illustrating MaxL statements and their components, in template form. In the diagrams, lowercase words are keywords, and uppercase words are intended to be replaced with the appropriate values, as shown in the example following each illustration.

Altering a Database

Figure 156 MaxL statement to change data file cache size

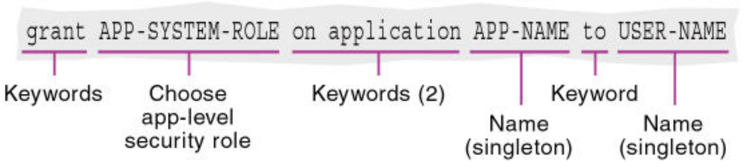

alter database DBS-NAME set data_file_cache_size SIZE-STRING
Keywords (2)
Name (double)
Keywords (2)
Positive number plus units

Example:

```
alter database Sample.Basic set data_file_cache_size 32768KB;
```

Granting a Permission

Figure 157 MaxL statement to grant application permissions to a user

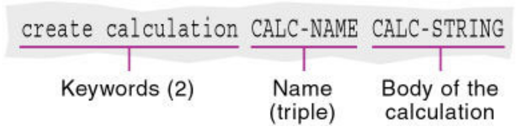

grant APP-SYSTEM-ROLE on application APP-NAME to USER-NAME
Keywords
Choose app-level security role
Keywords (2)
Name (singleton)
Keyword
Name (singleton)

Example:

```
grant designer on application Sample to Fiona;
```

Creating a Calculation

Figure 158 MaxL statement to create a stored calculation



Example:

```
create calculation sample.basic.varcalc
'"Variance"=@VAR(Actual, Budget);
"Variance %"=@VARPER(Actual, Budget);'
;
```

The MaxL Shell

This section shows you how to get started using most of the features of the MaxL Shell. Also see the *Oracle Essbase Technical Reference*.

This section does not discuss using the Administration Services MaxL Script Editor. See the *Oracle Essbase Administration Services Online Help*.

Starting the MaxL Shell

The MaxL Shell can be invoked to take input in these ways:

- Interactively, from the keyboard
- From a MaxL script file (statements are read from the file specified on the command line)
- From standard input that is piped to the MaxL Shell from the output of another program

The MaxL Shell also accepts command-line arguments at invocation time. These can be used with positional parameters to represent any name, or a password.

Starting the Shell for Interactive Input

In these examples, text that you enter is indicated in bold.

➤ To enter MaxL statements interactively at the command line, invoke the shell at your operating-system prompt.

For example:

```
essmsh
```

➤ To enter MaxL statements interactively after logging in at invocation time, use the `-l` flag.

For example:

```
essmsh -l Admin password
...
49 - User logged in: [Admin].
```

➤ To enter MaxL statements interactively and supply command-line arguments to represent variables that you will use in your interactive session, use the `-a` flag. For example:

```
essmsh -a Admin password Sample.Basic
...
login $1 $2;

49 - User logged in: [admin].

alter database $3.$4 enable aggregate_missing;

72 - Database altered: ['sample'.'basic'].
```

In the above example, `$1`, `$2`, `$3`, and `$4` are positional parameter variables representing arguments entered after `essmsh` at invocation time, in the order in which they were entered.

Starting the Shell for File Input

➤ To invoke the MaxL Shell to take input from a MaxL script file, type `essmsh` followed by the name of a MaxL script in the current directory, or, the full path and file name of a MaxL script in another directory.

If you provide only a file name, the MaxL Shell assumes that the file is in the current directory (the directory the operating-system command prompt was in when essmsh was invoked). In the following invocation example, the file `maxlscript.msh` must be in `C:\`.

```
C:\> essmsh maxlscript.msh
```

If the MaxL script is not in the current directory, provide a path to the MaxL script. You can use absolute paths or relative paths.

For example:

```
$ essmsh ../hyperion/products/Essbase/EssbaseServer/test.msh
```

Note:

MaxL scripts do not require a particular—or any—file extension. This document uses `.msh`.

In UNIX shells, place single quotation marks around the path to avoid file-reading errors.

In a Windows command prompt, if the path to the script contains a space, you may have to use double quotation marks around the entire path and file name to avoid file-reading errors.

Starting the Shell for Programmatic Input

➤ To invoke the MaxL Shell to take input from the standard output of another program or process, use the `-i` flag. For example:

```
program.sh | essmsh -i
```

The shell script `program.sh` may generate MaxL statements as output. The shell script output is piped to `essmsh -i`, which uses that output as its input. This allows for efficient co-execution of scripts.

Windows Example Using -i Invocation

The following Windows batch script generates a login statement and a MaxL display statement as its output. The `-i` flag enables that output to be used by `essmsh`, the MaxL Shell, as input.

```
echo login admin password on localhost; display privilege user;|essmsh -i
```

User Admin is logged in, all user privileges are displayed, and the MaxL session is terminated.

UNIX Example Using -i Invocation

The following portion of a shell script ensures that there are no applications on the system by testing whether the **display application** statement returns zero applications.

```
if [ $(echo "display application;" |
essmsh -l admin password -i 2>&1 |
awk '/Records returned/ {print $7}' ) != "[0]." ]
then
   print "This test requires that there be no applications."
   print "Quitting."
   exit 2
fi
```

Explanation:

1. MaxL grammar is piped to a MaxL Shell invocation and login, as the output of the UNIX `echo` command.
2. The results of the MaxL session are tested by `awk` for pattern-matching with the MaxL status message you would get if you entered **display application** on an empty system: `Records returned: [0].`

3. `Awk` matches the string `'Records returned: '`, and then checks to see if that is equal to `'[0].'`
4. If `$7` (a variable representing the fifth token `awk` finds in the status string) is equal to `'[0].'`, there are no applications on the system; otherwise, `$7` would equal `'[1].'` or whatever number of applications exist on the system.

For more information and examples on invocation options, see the *Oracle Essbase Technical Reference*. Invocation information is also contained in the essmsh "man page". To view the man page, enter `essmsh -h | more` at the operating-system command prompt.

Logging In to Essbase

The MaxL language interpreter requires a connection to an Essbase session before it can begin parsing MaxL statements. Use the MaxL Shell to establish the connection to Essbase.

➤ To log in to Essbase after the command shell has been started, use the shell **login** grammar. Text that you enter is indicated by bold text.

For example:

```
essmsh

MAXL>login Fiona identified by sunflower on hostname;
```

If a host name is not specified, `localhost` is assumed.

➤ To log in when you invoke the shell, use the `-l` option. To log in to a server besides `localhost` at invocation time, use the `-s` option. To set a message level, use `-m`. For example:

```
essmsh -l fiona sunflower -s myHost -m error
```

Note:

You can log out and change users without quitting the shell.

For more information about MaxL Shell invocation options, see the MaxL DDL > MaxL Shell section of the *Oracle Essbase Technical Reference*.

Command Shell Features

The MaxL Shell includes command-line argument processing, environment variable processing, nesting of MaxL scripts, and shell escapes. These features offer the flexibility needed to create a highly automated Essbase production environment.

See the *Oracle Essbase Technical Reference*.

Nesting MaxL Scripts

As a DBA, you may want to save your separate automated tasks in several MaxL scripts, rather than executing many operations from a single script. Putting the pieces together is simple if you know how to reference one MaxL script from another.

➤ To reference or include other MaxL scripts within the current MaxL session, use the following MaxL Shell syntax:

```
msh <scriptfile>;
```

Spooling Output to a File

You can create a log file of all or part of a MaxL session and its associated messages by spooling output to a file.

➤ To record a MaxL session:

1 **Log in to Essbase. For example,**

```
login fiona sunflower;
```

2 **Begin spooling output, using spool on to <filename>. For example:**

```
spool on to 'C:\\output\\display.txt';
```

3 **Enter as many MaxL statements as you want recorded.**

4 **Stop spooling output, using spool off;.**

MaxL statements and their output are logged to the output file when you issue the spool command, either in an interactive session or in a script. However, MaxL Shell commands and output are logged only if you spool during an interactive session. MaxL Shell commands and output are ignored in log files created from script sessions. Additionally, output from any operating-system commands you may have included is ignored in the log files of both interactive and script sessions.

Including Operating-System Commands

You can issue operating-system commands directly from a MaxL session. The operating-system output becomes part of the MaxL Shell output. When the operating system finishes executing commands, it returns control to `essmsh`.

➤ To escape to the operating system from within a MaxL session, use **shell**. For example, to run the UNIX **date** command from a MaxL script:

```
shell date;
```

➤ To escape to ESSCMD from within a MaxL session:

```
shell esscmd '../scripts/test.scr';
```

Using Variables to Customize MaxL Scripts

In the MaxL Shell, you can use variables as placeholders for data that is subject to change or that you refer to often; for example, the name of a computer, user names, or passwords. You can use variables in MaxL scripts and during interactive sessions. Using variables in MaxL scripts eliminates the need to create customized scripts for each user, database, or host. Variables can be environment variables (for example, `$ESSBASEPATH`, which references the Essbase installation directory), positional parameters (for example, `$1`, `$2`, and so on), and locally defined shell variables. A variable always begins with a `$` (dollar sign) when you reference it.

For more information about using variables in the MaxL Shell, see the *Oracle Essbase Technical Reference*.

Stopping the MaxL Shell

You can log out of a MaxL session or log in as another user without quitting the shell. You should include a logout statement at the end of MaxL scripts. You need not exit at the end of MaxL script files, or after a session using stream-oriented input from another program's output.

➤ To log out without exiting the MaxL Shell, enter:

```
logout;
```

➤ To exit from the MaxL Shell after using interactive mode, enter:

```
exit;
```

The MaxL Perl Module

With the MaxL Perl Module (`Essbase.pm`), you can embed the MaxL language within Perl programs, offering more programmatic control than is available in the shell.

`Essbase.pm`, in the `Perlmod` directory, enables Perl programmers to wrap MaxL statements in Perl scripts. Database administration with MaxL becomes as efficient and flexible as your Perl programs.

While you administer Essbase databases, Perl with MaxL enables you to take advantage of these and other programmatic features:

- Conditional testing
- Interprocess communication
- Message handling
- E-mail notification
- Web scripting

`Essbase.pm` contains methods that enable passing MaxL statements by means of Perl:

- **connect** () establishes a connection to Essbase
- **do** () tells Perl to execute the enclosed MaxL statement

- **pop_msg** () navigates through the stack of returned MaxL messages
- **fetch_col** (), **fetch_desc** (), and **fetch_row** () retrieve information from MaxL display output tables
- **disconnect** () terminates the connection to Essbase

To make the Perl methods available to Essbase, include a reference to `Essbase.pm` in the Perl program. Place the following line at the top of each Perl script:

```
use Essbase;
```

Perl is not difficult to learn, especially if you have knowledge of UNIX shells or other programming languages. To download Perl and learn more about it, visit the Comprehensive Perl Archive Network Web site at http://www.cpan.org/.

See the MaxL DDL section of the *Oracle Essbase Technical Reference* and the README file in the PERLMOD directory of your Essbase installation.

Part IX

Optimizing Essbase

In Optimizing Essbase:

50 Monitoring Performance

In This Chapter

Also see:

- "Using Essbase Logs" on page 729
- *Oracle Essbase Error Message Reference*

Finding Diagnostic Information

Essbase provides performance information dialog boxes at the Essbase Server, application, and database level. View performance information before performing these tasks:

- Preparing to migrate data
- Adding users
- Analyzing performance problems
- Performing other administrative tasks

Essbase displays information on a snapshot basis; to see the latest information, click Refresh. If the refresh button is displayed in a window or dialog box, it updates every tab in the window or dialog box.

For a comprehensive discussion of Essbase Server, application, and outline logs, see "Essbase Server and Application Logs" on page 730.

The following sections provide procedures for accessing information about Essbase Servers, applications, and databases that are commonly used to diagnose performance or other issues.

Viewing Essbase Server Information

You can view information about the Essbase Server license, configuration, operating system, disk drives, and applications for Essbase Server.

➤ To view Essbase Server information, use a tool:

Tool	Topic	Location
Administration Services	Monitoring Essbase Servers	*Oracle Essbase Administration Services Online Help*
MaxL	**display application**	*Oracle Essbase Technical Reference*
ESSCMD	GETAPPSTATE	*Oracle Essbase Technical Reference*

Viewing Application Information

You can view application information to identify which databases are running in the application and to check access, security, and startup information.

➤ To view application information, use a tool:

Tool	Topic	Location
Administration Services	Monitoring Applications	*Oracle Essbase Administration Services Online Help*
MaxL	**display application**	*Oracle Essbase Technical Reference*
ESSCMD	GETAPPSTATE	*Oracle Essbase Technical Reference*

Viewing Database Information

You can view information about database storage, database statistics, and lock contention. This information may help you identify activities or operations that affect performance.

➤ To view database information, use a tool:

Tool	Instructions	For More Information
Administration Services	Monitoring Databases	*Oracle Essbase Administration Services Online Help*
MaxL	**display database**	*Oracle Essbase Technical Reference*
ESSCMD	● For general information: GETDBINFO ● For database storage information: GETDBSTATE	*Oracle Essbase Technical Reference*

Tool	Instructions	For More Information
	● For currency information: GETCRDBINFO ● For database statistics: GETDBSTATS ● For runtime information: GETDBINFO	

Monitoring the Status of Applications and Databases

You can view the start and stop status of applications and databases that you are authorized to use.

➤ To view application/database status, use a tool:

Tool	Instructions	For More Information
Administration Services	Viewing Application/Database Status	*Oracle Essbase Administration Services Online Help*
ESSCMD	GETAPPINFO GETDBINFO	*Oracle Essbase Technical Reference*

Monitoring User Sessions and Requests

You can monitor active user sessions for an Essbase Server, application, or database. If you have Administrator or Application Manager permissions, you can disconnect a user session or terminate a request made during a session. See "Disconnecting Users and Terminating Requests in Native Security Mode" on page 639.

➤ To monitor user sessions and requests, use a tool:

Tool	Instructions	For More Information
Administration Services	Viewing Active User Sessions	*Oracle Essbase Administration Services Online Help*
MaxL	**display session alter system**	*Oracle Essbase Technical Reference*

Monitoring Applications from the Operating System

Each application that is loaded is an open task or process in the operating system. You can use the operating system to view application tasks or processes:

- On Windows platforms, the application is displayed in an application server window, where you can view application activities as they occur. When the Agent starts an application, a new icon is displayed in the taskbar. Double-click the icon to view the application server window.

- On UNIX platforms, the application server is often a background process. When the application starts, the ESSBASE command starts the ESSSVR process. To see activities, you can route all messages to a file with the `tail -f` *`log`* command, where *`log`* is the name of a file that you specify.
- You can also view a snapshot of the Essbase Server or application log using Administration Services. For information about viewing, filtering, searching, and analyzing logs, see *Oracle Essbase Administration Services Online Help*. For information on server and application logs, see "Essbase Server and Application Logs" on page 730.

51 Improving Essbase Performance

In This Chapter

Recognizing Basic Design Issues That Affect Optimization

Use the following list to identify basic design issues that affect optimization outside this volume:

- For an introduction to basic concepts and how they relate to optimized performance, see Chapter 3, "Understanding Multidimensional Databases."
- For a discussion of how to analyze a database design while it is still on paper and of how this analysis can aid optimization, see "Analyzing and Planning" on page 81.
- To understand how basic database outline issues affect performance, see "Designing an Outline to Optimize Performance" on page 95.

Resetting Databases to Improve Performance

You can periodically reset a database and then reload it. Even if you reload a database very often, the main database (`.pag`) files can grow unless you reset the database.

➤ To reset a database, use a tool:

Tool	Topic	Location
MaxL	**alter database *appname.dbname* reset**	*Oracle Essbase Technical Reference*
ESSCMD	RESETDB	*Oracle Essbase Technical Reference*

Using Database Settings to Customize for Maximum Performance

You can customize Essbase for maximum performance, using database settings at the database level in Administration Services, ESSCMD, or MaxL.

Note:

If you are migrating a database, see the *Oracle Hyperion Enterprise Performance Management System Installation and Configuration Guide* for information about the default settings after migration.

The following sections list performance settings and describe how to adjust them.

Database Cache Settings

Table 98 describes database cache settings and lists the location of the settings in Administration Services, MaxL, and ESSCMD:

Table 98 Database Cache Settings

Setting	More Information	Location in Administration Services, MaxL, ESSCMD
Index cache size	"Sizing the Index Cache" on page 827	● Administration Services: Database Properties window, Caches tab ● MaxL: **alter database** ***appname.dbname*** **set index_cache_size** ***n*** ● ESSCMD: SETDBSTATEITEM 12
Data file cache size	"Sizing the Data File Cache" on page 828	● Administration Services: Database Properties window, Caches tab ● MaxL: **alter database** ***appname.dbname*** **set data_file_cache_size** ***n*** ● ESSCMD: SETDBSTATEITEM 27
Data cache size	"Sizing the Data Cache" on page 829	● Administration Services: Database Properties window, Caches tab ● MaxL: **alter database** ***appname.dbname*** **set data_cache_size** ***n*** ● ESSCMD: SETDBSTATEITEM 5
Index page size	Fixed size	N/A

Setting	More Information	Location in Administration Services, MaxL, ESSCMD
Cache memory locking	"Deciding Whether to Use Cache Memory Locking" on page 826	● Administration Services: Database Properties window, Caches tab ● MaxL: **alter database *appname.dbname* enable cache_pinning** ● ESSCMD: SETDBSTATEITEM 26

See "Setting Database Properties" in *Oracle Essbase Administration Services Online Help*.

Database Disk Volumes Settings

Table 99 describes database disk volume settings and lists the location of the settings in Administration Services, MaxL, and ESSCMD:

Table 99 Database Disk Volume Settings

Setting	More Information	Location in Administration Services, MaxL, ESSCMD
Volume name	"Specifying Disk Volumes" on page 768	● Administration Services: Database Properties window, Storage tab ● MaxL: **alter database *appname.dbname* set disk volume** ● ESSCMD: SETDBSTATEITEM 23 SETDBSTATEITEM 24
Partition size	"Specifying Disk Volumes" on page 768	● Administration Services: Database Properties window, Storage tab ● MaxL: **alter database *appname.dbname* set disk volume** ● ESSCMD: SETDBSTATEITEM 23 SETDBSTATEITEM 24
File type	"Specifying Disk Volumes" on page 768	● Administration Services: Database Properties window, Storage tab ● MaxL: **alter database *appname.dbname* set disk volume** ● ESSCMD: SETDBSTATEITEM 23
Maximum file size	"Specifying Disk Volumes" on page 768	● Administration Services: Database Properties window, Storage tab ● MaxL: **alter database *appname.dbname* set disk volume** ● ESSCMD: SETDBSTATEITEM 23

See "Setting Disk Volumes" in *Oracle Essbase Administration Services Online Help*.

Database Transaction Control Settings

Table 100 describes database transaction control settings and lists the location of the settings in Administration Services, MaxL, and ESSCMD:

Table 100 Database Transaction Control Settings

Setting	More Information	Location in Administration Services, MaxL, ESSCMD
Isolation level	"Understanding Isolation Levels" on page 779	● Administration Services: Database Properties window, Transactions tab ● MaxL: **alter database *appname.dbname* enable committed_mode** ● ESSCMD: SETDBSTATEITEM 18
Commit Blocks	"Understanding Isolation Levels" on page 779	● Administration Services: Database Properties window, Transactions tab ● MaxL: **alter database *appname.dbname* enable committed_mode** and **alter database *appname.dbname* set implicit_commit after *n* blocks** ● ESSCMD: SETDBSTATEITEM 21
Commit Rows	"Understanding Isolation Levels" on page 779	● Administration Services: Database Properties window, Transactions tab ● MaxL: **alter database *appname.dbname* enable committed_mode** and **alter database *appname.dbname* set implicit_commit after *n* rows** ● ESSCMD: SETDBSTATEITEM 22
Wait for write access to locked data block	"Understanding Isolation Levels" on page 779	● Administration Services: Database Properties window, Transactions tab ● MaxL: **alter database *appname.dbname* set lock_timeout** ● ESSCMD: SETDBSTATEITEM 20
Pre-image access	"Understanding Isolation Levels" on page 779	● Administration Services: Database Properties window, Transactions tab ● MaxL:

Setting	More Information	Location in Administration Services, MaxL, ESSCMD
		alter database ***appname.dbname*** **enable pre_image_access** ● ESSCMD: SETDBSTATEITEM 19

See “Setting Data Integrity Options” in *Oracle Essbase Administration Services Online Help*.

Miscellaneous Database Settings

Table 101 describes miscellaneous database settings and lists the location of the settings in Administration Services, MaxL, and ESSCMD:

Table 101 Miscellaneous Database Settings

Setting	More Information	Location inAdministration Services, MaxL, ESSCMD
Retrieval buffer size	“Setting the Retrieval Buffer Size” on page 899	● Administration Services: Database Properties window, General tab ● MaxL: **alter database** ***appname.dbname*** **set retrieve_buffer_size** ***n*** ● ESSCMD SETDBSTATEITEM 16
Retrieval sort buffer size	“Setting the Retrieval Sort Buffer Size” on page 900	● Administration Services: Database Properties window, General tab ● MaxL: **alter database** ***appname.dbname*** **set retrieve_sort_ buffer_size** ***n*** ● ESSCMD: SETDBSTATEITEM 17
Data compression	“Data Compression” on page 772	● Administration Services: Database Properties window, Storage tab ● MaxL: **alter database** ***appname.dbname*** **enable compression** and **alter database** ***appname.dbname*** **set compression** ***type*** ● ESSCMD: SETDBSTATEITEM 14 SETDBSTATEITEM 15
Maximum memory for trigger definitions	“Understanding Triggers Definitions” on page 119	MaxL: **create or replace trigger**, **alter trigger display trigger**, and **drop trigger**

See “Setting Database Properties” in *Oracle Essbase Administration Services Online Help*.

Eliminating Fragmentation

Fragmentation is created when Essbase writes a data block to a new location on disk and leaves unused space in the former location of the data block. Block size increases because data from a data load or calculation is appended to the blocks; the blocks must therefore be written to the end of a data file.

The Essbase Kernel merges adjacent fragments into increasingly larger fragments so that unused space more likely will be reused.

In some cases, fragmentation cannot be reduced completely. Fragmentation is likely to occur in the following situations:

- Read/write databases that users are constantly updating with data
- Databases that execute calculations around the clock
- Databases that frequently update and recalculate dense members
- Data loads that are poorly designed
- Databases that contain a significant number of Dynamic Calc and Store members
- Databases that use an isolation level of uncommitted access with commit block set to zero

If you experience performance slowdowns, check to see if there is too much fragmentation of the database; if there is, you can take steps to reduce it.

Measuring Fragmentation

You can measure fragmentation using the average clustering ratio or average fragmentation quotient statistic:

Using the Average Fragmentation Quotient

In ESSCMD, look at the Average Fragmentation Quotient that is returned when you execute the GETDBSTATS command. Use this table to evaluate whether the level likely will cause performance problems:

Database Size	Fragmentation Quotient Threshold
Small (<200 MB)	60% or greater
Medium (<2 GB)	40% or greater
Large (>2 GB)	30% or greater

Any quotient above the high end of the range indicates that reducing fragmentation may help performance, with the following qualifications:

- The reported value of the Fragmentation Quotient is more accurate when no other write transactions run on the database.
- For databases less than 50 MB using the Direct I/O access mode, the fragmentation quotient tends to be high. A high fragmentation quotient does not necessarily indicate a need to reduce fragmentation, because the free space is created in 8 MB chunks, and all of it might not get used immediately.

Using the Average Clustering Ratio

The average clustering ratio database statistic indicates the fragmentation level of the data (`.pag`) files. The maximum value, 1, indicates no fragmentation.

➤ To view the average clustering ratio for a database, use a tool:

Tool	Topic	Location
Administration Services	Viewing Fragmentation Statistics	*Oracle Essbase Administration Services Online Help*
ESSCMD	GETDBSTATS	*Oracle Essbase Technical Reference*

Preventing or Removing Fragmentation

You can prevent and remove fragmentation:

- To prevent fragmentation, optimize data loads by sorting load records based on sparse dimension members. For a comprehensive discussion of optimizing data load by grouping sparse members, see "Grouping Sparse Member Combinations" on page 852.
- To remove fragmentation, perform an export of the database, delete all data in the database with CLEARDATA, and reload the export file. See the *Oracle Hyperion Enterprise Performance Management System Backup and Recovery Guide*.
- To remove fragmentation, force a dense restructure of the database. See "Types of Database Restructuring" on page 842.

Enabling Windows 4 GB RAM Tuning

Essbase supports Microsoft 4 GB RAM Tuning (4GT). 4GT enables users with extremely large databases to take advantage of a larger address space to improve performance.

The total addressable limit of RAM on servers running Windows 2000 Server is 4 GB. By default, applications can access 2 GB, with the Windows kernel using the other 2 GB. For selected versions of Windows running on Intel architecture servers, Microsoft provides the 4GT feature. The 4GT feature increases the addressable limit for applications to 3 GB, reducing the potential RAM allocated to the Windows kernel from 2 GB to 1 GB.

Essbase currently supports the 4GT feature on computers that use Intel-based processors with more than 2 GB of physical RAM and are running Windows 2000 Server.

Enabling the Windows 4GT feature may benefit users if the Essbase installation has the following characteristics:

- Essbase is configured to use direct I/O. Enabling 4GT on Essbase installations configured for buffered I/O is not recommended.
- The index and data caches are sized correctly, and Essbase performance is consistently improved by increasing the data file cache, but further increases are bounded by the previous 2 GB addressability limitation. For information about setting cache values, see Chapter 52, "Optimizing Essbase Caches."

To enable the 4GT feature on a computer where Essbase is installed, modify the `boot.ini` file by adding `/3GB` to the startup line for each boot partition that is defined for a Windows version that supports 4GT; for example:

```
[boot loader]
timeout=30
default=multi(0)disk(0)rdisk(0)partition(2)\WIN2KADV
[operating systems]
multi(0)disk(0)rdisk(0)partition(2)\WIN2KADV="Microsoft Windows 2000
Advanced Server" /3GB
```

Note:

This change to the `boot.ini` file is effective only if you are using Windows 2000 Advanced Server. On a dual boot system, to ensure that the boot loader supporting the 4GT feature is installed, ensure that the last operating system you install is a version that supports the 4GT feature. Because the Windows kernel area is reduced, it is unlikely but conceivable that certain applications and workloads in certain environments may experience degraded performance. Testing your workload in your environment is recommended.

For additional information about the Microsoft Windows 4GT feature, see http:\\www.microsoft.com.

Implementing 64-bit Essbase

Because processes on 64-bit have greatly increased memory addressability over 32-bit processes, the 64-bit edition of Essbase can handle larger outlines and cache sizes than 32-bit Essbase. In computing environments that support it, implementing 64-bit Essbase can improve the performance of existing applications and can sustain much larger applications. For information on platform support for 64-bit Essbase, see the *Oracle Hyperion Enterprise Performance Management System Installation Start Here*.

On 64-bit Essbase, you can set cache sizes larger than the existing 32-bit limits. In Essbase clients, the maximum values you can set for the data cache, data file cache, and index cache is 4 GB. You can enable larger values for the data cache and data file cache using the MEMSCALINGFACTOR configuration setting. See the *Oracle Essbase Technical Reference*.

The maximum thread settings are higher for 64-bit Essbase than for 32-bit Essbase. Table 102 lists the maximum thread settings for each. For information about changing thread settings, see the *Oracle Essbase Technical Reference*. For information about multi-threading, see "Multithreading" on page 688.

Table 102 Maximum Thread Settings for 32-bit and 64-bit Essbase

Setting	32-bit Maximum	64-bit Maximum
AGENTTHREADS	500	1024
SERVERTHREADS	500	1024
DLTHREADSPREPARE	16	32
DLTHREADSWRITE	16	32

Default retrieval buffer settings for 64-bit Essbase are higher than for 32-bit Essbase. Table 103 lists the default retrieval buffer settings for each. See "Changing Buffer Size" on page 899.

Table 103 Default Retrieval Buffer Settings for 64-bit Essbase

Setting	32-bit Default	64-bit Default
Retrieval Buffer	10 KB	20 KB
Retrieval Sort Buffer	10 KB	20 KB

Note:

Because of internal data structure size changes, 64-bit Essbase requires a larger retrieval sort buffer size than 32-bit Essbase. If you encounter the error, `"Sort buffer limit of [x] rows have been exceeded"` (where *x* is the current maximum number of rows allowed for the current buffer size), increase the retrieval sort buffer size by a factor of two.

Finding Additional Optimization Information

"Using Database Settings to Customize for Maximum Performance" on page 816 provides general-purpose information and does not account for the wide variety of configuration possibilities. For more information about performance and server, application, or other settings, see these chapters:

- For information on optimizing performance for a particular function, see these chapters:
  - Chapter 54, "Optimizing Data Loads"
  - Chapter 55, "Optimizing Calculations"
  - Chapter 25, "Understanding Intelligent Calculation"
  - Chapter 56, "Optimizing Reports and Other Types of Retrieval"
- For information on the Essbase Kernel, see the following chapters:
  - Chapter 46, "Managing Database Settings"

- Chapter 47, “Allocating Storage and Compressing Data”
- Chapter 48, “Ensuring Data Integrity”
- Chapter 52, “Optimizing Essbase Caches”
- Appendix D, “Estimating Disk and Memory Requirements”

52 Optimizing Essbase Caches

In This Chapter

The information in this chapter applies only to block storage databases and is not relevant to aggregate storage databases. Also see Chapter 57, "Comparison of Aggregate and Block Storage."

Understanding Essbase Caches

Essbase uses five memory caches to coordinate memory usage:

Table 104 Essbase Caches

Cache	Description
Index cache	A buffer in memory that holds index pages. How many index pages are in memory at one time depends upon the amount of memory allocated to the cache.
Data file cache	A buffer in memory that holds compressed data files (`.pag` files). Essbase allocates memory to the data file cache during data load, calculation, and retrieval operations, as needed. The data file cache is used only when direct I/O is in effect.
Data cache	A buffer in memory that holds uncompressed data blocks. Essbase allocates memory to the data cache during data load, calculation, and retrieval operations, as needed.
Calculator cache	A buffer in memory that Essbase uses to create and track data blocks during calculation operations.
Dynamic calculator cache	A buffer in memory that Essbase uses to store all of the blocks needed for a calculation of a Dynamic Calc member in a dense dimension (for example, for a query).

Essbase provides default size settings for each cache; you can adjust the sizes as needed for each database. Appropriate cache size is affected by many factors, including database size, block size, index size, and available server memory. Cache size settings can significantly affect database and general server performance.

The following topics provide information about sizing caches for performance.

Deciding Whether to Use Cache Memory Locking

Before setting cache sizes, you must enable cache memory locking or leave cache memory locking disabled (the default).

The setting for cache memory locking controls whether the memory used for the index cache, data file cache, and data cache are locked into physical memory, giving the Essbase kernel priority use of system RAM.

To use cache memory locking, you must be using direct I/O (buffered I/O is the default I/O access mode), and direct I/O requires a larger index cache size than buffered I/O. See Chapter 46, "Managing Database Settings."

Locking improves performance for an Essbase database because the system memory manager need not swap the memory used by the caches when swapping the memory used by Essbase Server. By default, cache memory locking is turned off.

Enabling cache memory locking gives the Essbase Kernel priority use of system RAM. If you enable cache memory locking, leave at least one-third of the system RAM available for non-Essbase Kernel use. If you do not want to give the Essbase Kernel priority usage of system RAM, do not enable cache memory locking.

If you are running Essbase on Solaris, run the Bourne shell script, `root.sh`, before starting Essbase and enabling cache memory locking. This script sets the server to run in Superuser mode so that it can lock memory. See the *Oracle Hyperion Enterprise Performance Management System Installation and Configuration Guide*.

➤ To enable cache memory locking, use a tool:

Tool	Topic	Location
Administration Services	Enabling Cache Memory Locking	*Oracle Essbase Administration Services Online Help*
MaxL	**alter database enable cache_pinning**	*Oracle Essbase Technical Reference*
ESSCMD	SETDBSTATEITEM 26	*Oracle Essbase Technical Reference*

Sizing Caches

The settings that you should use for each of the caches that you can configure depend on data distribution and the dense/sparse configuration of the database. The maximum combined total size of the caches should equal the amount of available memory after the memory required by Essbase is taken into account.

The needs for each site, even for a particular database, can vary. Depending on the complexity and type of each operation, Essbase allocates as much memory for the data file cache and the data cache as needed. Use the recommended values in this section to estimate enough memory for *optimal* performance.

If you are using Essbase for the first time, cache sizes are set automatically to the default values discussed in the following sections. Use these topics to find and understand recommendations for each cache size.

Note:

Changes made to cache sizes take effect the next time you start the database.

Sizing the Index Cache

The index is stored in index files on disk. When a database is active, the most recently accessed index pages are held in the index cache. How much of the index can be held in memory at one time depends on the amount of memory that you allocate to the index cache.

Note:

The size of index pages is fixed at 8 K to reduce input-output overhead, as well as to simplify database migration.

The effectiveness of the index cache size depends on the nature of the calculation. For example, if you were reloading and recalculating an entire database (such as a database that is refreshed each month), a high index cache size is not helpful, because Essbase is creating blocks rather than searching the index cache for existing blocks during calculation.

Table 105 shows default and recommended settings for the index cache:

Table 105 Index Cache Size Settings

Minimum Value	Default Value	Recommended Value
1024 KB (1,048,576 bytes)	Buffered I/O: 1024 KB (1,048,576 bytes) Direct I/O: 10,240 KB (10,485,760 bytes)	Combined size of all `ess`*`n`*`.ind` files, if possible; otherwise, as large as possible. Do not set this cache size higher than the total index size, because no performance improvement results. To determine the total index size, see "Index Files" on page 1028.

For information about changing the I/O access mode for a database, or about changing the default for all newly created databases, see "Understanding Buffered I/O and Direct I/O" on page 758.

In general, if you are using direct I/O, make the index cache as large as system resources allow. If you are using buffered I/O, make the index cache as small as possible.

For information on testing and fine-tuning cache settings, see "Fine-Tuning Cache Settings" on page 838.

Changing the Index Cache Size

➤ To set the size of the index cache, use a tool:

Tool	Topic	Location
Administration Services	Setting Cache Sizes	*Oracle Essbase Administration Services Online Help*
MaxL	**alter database set index_cache_size**	*Oracle Essbase Technical Reference*
ESSCMD	SETDBSTATEITEM 12 SETDBSTATE	*Oracle Essbase Technical Reference*

Sizing the Data File Cache

The data file cache holds data files (`.pag` files) in memory if you are using direct I/O. If you are not using direct I/O, the data file cache is not used. How much of the data within data files can fit into memory at one time depends on the amount of memory you allocate to the data file cache.

In general, if you must choose whether to allocate memory to the data cache or to the data file cache, choose the data file cache if you are using direct I/O.

Table 106 shows default and recommended settings for the data file cache:

Table 106 Data File Cache Size Settings

Minimum Value	Default Value	Recommended Value
Direct I/O: 10,240 KB (10,485,760 bytes)	Direct I/O: 32,768 KB (33,554,432 bytes)	Combined size of all `ess`*`n`*`.pag` files, if possible; otherwise, as large as possible. This cache setting not used if Essbase is set to use buffered I/O.

In general, if you are using direct I/O, make the data file cache as large as system resources allow. If you are using buffered I/O, the data file cache is not used.

For information on testing and fine-tuning cache settings, see "Fine-Tuning Cache Settings" on page 838.

Changing the Data File Cache Size

➤ To set the size of the data file cache, use a tool:

Tool	Topic	Location
Administration Services	Setting Cache Sizes	*Oracle Essbase Administration Services Online Help*
MaxL	**alter database set data_file_cache_size**	*Oracle Essbase Technical Reference*
ESSCMD	SETDBSTATEITEM 27	*Oracle Essbase Technical Reference*

Sizing the Data Cache

Data blocks reside on physical disk and in memory. The number of blocks that can be held in the data cache at one time depends on how much memory you allocate to the data cache.

When a block is requested, Essbase searches the data cache for the block. If Essbase finds the block in the cache, it is accessed immediately. If the block is not found in the cache, Essbase searches the index for the appropriate block number and then uses the index entry of the block to retrieve it from the appropriate data file on disk. Retrieving a requested block from the data cache is faster and therefore improves performance.

In general, if you must choose whether to allocate memory to the data cache or to the data file cache, choose the data file cache if you are using direct I/O.

Table 107 shows default and recommended settings for the data cache:

Table 107 Data Cache Size Settings

Minimum Value	Default Value	Recommended Value
3072 KB (3145728 bytes)	3072 KB (3,145,728 bytes)	0.125 * the value of data file cache size. Increase value if any of these conditions exist: ● Many concurrent users are accessing different data blocks. ● Calculation scripts contain functions on sparse ranges, and the functions require all members of a range to be in memory (for example, when using @RANK and @RANGE). ● For data load, the number of threads specified by the DLTHREADSWRITE setting is high and the expanded block size is large.

In certain cases, you might need to increase the data cache when running concurrent calculations; for example, when concurrent calculations do not share any common blocks and the sparse member with the largest number of children has all its children blocks populated in the database. To calculate the data cache for concurrent calculations, use the following formula:

```
(Size of big block in bytes) * max(Number of children for a Sparse member)
* (Number of concurrent batch calc processes) * 2
```

If other concurrent operations (such as data loads and queries) take place when running concurrent calculations, increase the data cache even more to accommodate these requests.

Make the data cache as small as possible whether you are using buffered I/O or direct I/O.

For information on testing and fine-tuning cache settings, see "Fine-Tuning Cache Settings" on page 838.

Note:

When running Essbase on 64-bit platforms, optimal data cache and data file cache settings may be larger than 4 GB. Although you cannot specify settings larger than 4 GB in Essbase clients, you can enable larger settings using the MEMSCALINGFACTOR configuration setting. See the *Oracle Essbase Technical Reference*.

Changing the Data Cache Size

➤ To set the size of the data cache, use a tool:

Tool	Topic	Location
Administration Services	Setting Cache Sizes	*Oracle Essbase Administration Services Online Help*
MaxL	**alter database set data_cache_size**	*Oracle Essbase Technical Reference*
ESSCMD	SETDBSTATEITEM 5 SETDBSTATE	*Oracle Essbase Technical Reference*

Sizing the Calculator Cache

Essbase can create a bitmap, whose size is controlled by the size of the calculator cache, to record and track data blocks during a calculation. Determining which blocks exist using the bitmap is faster than accessing the disk to obtain the information, particularly if calculating a database for the first time or calculating a database when the data is sparse.

Essbase uses the calculator cache bitmap if the database has at least two sparse dimensions and either of these conditions is also met:

- You calculate at least one full sparse dimension.
- You specify the SET CACHE ALL command in a calculation script.

The best size for the calculator cache depends on the number and density of the sparse dimensions in your outline. Use the following topics to understand the calculator cache bitmap, size the calculator cache, and change the size of the calculator cache (and therefore the largest possible size for the bitmap), if required:

Understanding the Calculator Cache Bitmap

For the calculator cache, Essbase separates sparse dimensions in the database into two groups:

- Bitmap dimensions: The sparse dimensions from the database outline that Essbase fits into the bitmap until the bitmap is full. Each member combination of the sparse dimensions placed in the bitmap occupies 1 bit of memory, and there must be enough space in the bitmap for every member combination of a sparse dimension for it to be placed in the bitmap.
- Anchoring dimensions: The remaining one or more sparse dimensions in the database outline that do not fit into the bitmap.

Essbase starts with the first sparse dimension in the database outline and fits as many sparse dimensions as possible into the bitmap. The dimensions that fit are the bitmap dimensions. Essbase stops the process when it cannot fit another complete sparse dimension into the bitmap. Because the calculator cache controls the size of the bitmap, the number of sparse dimensions that can fit in the bitmap depends on the size of the calculator cache (and the number and size of the sparse dimensions).

The remaining sparse dimensions are the anchoring dimensions. For anchoring dimensions, Essbase cannot use the bitmap to determine whether blocks exist.

To see which dimensions are anchoring dimensions and which are bitmap dimensions, use the SET MSG DETAIL calculation command to display bitmap information in the application log.

Carefully order the sparse dimensions in your outline so that as many dimensions as possible can be placed into the bitmap. Start with the dimension that contains the fewest members and continue until the dimension with the most members is last. This order allows more dimensions to fit into the bitmap and results in improved calculation performance.

Note:

The order of sparse dimensions in the outline also affects query performance. See “Optimizing Query Performance” on page 95.

Essbase uses a single bitmap if there are multiple anchoring dimensions or if the calculator cache is not large enough to support multiple bitmaps, and uses multiple bitmaps if there is one anchoring dimension.

A single bitmap has these properties:

- A single bitmap is used to track child blocks.
- A single bitmap uses the least memory but is less efficient than multiple bitmaps.

Multiple bitmaps have these properties:

- Multiple bitmaps are used, one to track child blocks and one to track parent blocks.
- Multiple bitmaps use more memory but are faster than using a single bitmap. The performance improvement is particularly high when you are calculating the database for the first time.

- The number of bitmaps used is determined by the maximum number of dependent parents for any members in the anchoring dimension. A member has one dependent parent, unless it has a shared member. For example, consider the Product dimension of the Sample.Basic database. The member Cola (100-10) has one parent, Colas (100). However, Diet Cola (100-20) has two parents, Diet Drinks (Diet) and Colas (100). No members of Product have more than two dependent parents. Therefore, if Product is the anchoring dimension, the maximum dependent parents is two.

Essbase chooses one of three options for the calculation:

Table 108 Options for calculator cache

Option	Method	Performance Rating
1	Single anchoring dimension, multiple bitmaps	1
2	Single anchoring dimension, single bitmap	2
3	Multiple anchoring dimensions, single bitmap	3

Essbase chooses the optimal performance method for a database calculation, based on the size of the calculator cache. If the calculator cache size is too small for any of the above options, Essbase does not use a calculator cache. Calculation performance may be significantly impaired.

Enabling parallel calculation may change which calculator cache option is used. See "Calculator Cache" on page 869.

Caution!

If you are calculating the database for the first time, the size of the calculator cache is particularly significant for calculation performance. If possible, ensure that the calculator cache is large enough for Essbase to use the optimal calculator cache option.

Calculating the Calculator Cache Size

The optimum size of the calculator cache depends on the memory that the system has available and on the nature and configuration of the database.

Using the following formula, you can calculate the calculator cache size required for Essbase to choose each of the three options in Table 108 on page 832.

Calculator cache	=	Bitmap size in bytes * Number of bitmaps
Bitmap size in bytes	=	Max ((member combinations on the bitmap dimensions / 8), 4)
Number of bitmaps	=	Maximum number of dependent parents in the anchoring dimension + 2 constant bitmaps

Note:

The minimum bitmap size is 4 bytes. If (member combinations on the bitmap dimensions/8) is less than 4 bytes, Essbase uses a bitmap size of 4 bytes.

Consider a sample database with five sparse dimensions (S1 to S5):

Sparse Dimension	Number of Members	Dependent Parents
S1	20	Not applicable
S2	20	Not applicable
S3	50	Not applicable
S4	50	Not applicable
S5	200	3

Use this sample database for the following sample calculations.

Option 1: Single Anchoring Dimension, Multiple Bitmaps

For this sample calculation, assume the following facts about a database (from Table 108 on page 832):

- Bitmap dimensions: S1, S2, S3, S4
- Anchoring dimension: S5
- Dependent parents in anchoring dimension: 3

Perform this calculation:

Bitmap size in bytes	=	(S1 * S2 * S3 * S4) / 8
	=	(20 * 20 * 50 * 50 / 8
	=	125,000 bytes
Number of bitmaps	=	Maximum number of dependent parents in the anchoring dimension
		+
	=	2 constant bitmaps
	=	3 + 2
		5
Calculator cache	=	Bitmap size * Number of bitmaps
	=	125,000 * 5
	=	**625,000 bytes**

For Essbase to use multiple bitmaps for this database with one anchoring dimension, the calculator cache must be 625,000 bytes.

Option 2: Single Anchoring Dimension, Single Bitmap

For this sample calculation, assume the following facts about a database (from Table 108 on page 832):

- Bitmap dimensions: S1, S2, S3, S4
- Anchoring dimension: S5
- Dependent parents in anchoring dimension: Not applicable

Perform this calculation:

Bitmap size in bytes	=	(S1 * S2 * S3 * S4) / 8
	=	(20 * 20 * 50 * 50) / 8
	=	125,000 bytes
Number of bitmaps	=	Single bitmap
	=	1
Calculator cache	=	Bitmap size * Number of bitmaps
	=	125,000 * 1
	=	**125,000 bytes**

For Essbase to use a single bitmap for this database with one anchoring dimension, the calculator cache must be 125,000 bytes.

Option 3: Multiple Anchoring Dimensions, Single Bitmap

For this sample calculation, assume the following facts about a database (from Table 108 on page 832):

- Bitmap dimensions: S1, S2, S3
- Anchoring dimensions: S4, S5
- Dependent parents in anchoring dimensions: Not applicable

Perform this calculation:

Bitmap size in bytes	=	(S1 * S2 * S3) / 8
	=	(20 * 20 * 50) / 8
	=	2,500 bytes
Number of bitmaps	=	Single bitmap
	=	1
Calculator cache	=	Bitmap size * Number of bitmaps
	=	2,500 * 1
	=	**2,500 bytes**

For Essbase to use a single bitmap for this database with multiple anchoring dimensions, the calculator cache must be 2,500 bytes.

Choosing a Calculator Cache Size for a Database

The following table shows which calculator cache option Essbase uses, depending on the calculator cache size specified:

Minimum Size Specified	Option Selected
625,000 bytes	Option 1 (provides optimal performance)
125,000 bytes	Option 2
2,500 bytes	Option 3

If you specify a calculator cache size of less than 2,500 bytes, Essbase does not use a calculator cache during the calculation. Calculation performance may be significantly impaired.

You can check which calculator cache option Essbase is able to use on a database by using the SET MSG SUMMARY command in a calculation script. Run the following calculation script on the empty database:

```
SET MSG SUMMARY;
CALC ALL;
```

Essbase displays the calculator cache setting in the ESSCMD window or in the application log. See "SET MSG SUMMARY and SET MSG DETAIL" on page 861.

The maximum calculator cache size that you can specify is 200,000,000 bytes. The default is 200,000 bytes. The calculator cache size that you choose depends on the memory is available and the configuration of the database.

Note:

The sizes of the calculator, index, data file, and data caches usually have a greater effect on performance if the database calculation is based more on aggregations and less on formula calculations.

Sizing the Calculator Cache to Calculate the Database for the First Time

If you are calculating the database for the first time, the size of the calculator cache is particularly significant. If possible, ensure that the calculator cache is large enough for Essbase to use the optimal calculator cache option. See "Calculating the Calculator Cache Size" on page 832.

Changing the Calculator Cache with Calculation Scripts

You can use the default calculator cache size, or you can set the size of the calculator cache within a calculation script. If you set the size from a calculation script, the setting is used only for the duration of the calculation script. See the calculation script SET CACHE command and CALCCACHE configuration setting in the *Oracle Essbase Technical Reference*.

Sizing Dynamic Calculator Caches

Essbase uses a separate dynamic calculator cache for each open database. The DYNCALCCACHEMAXSIZE setting in the `essbase.cfg` file specifies the maximum size of each dynamic calculator cache on the server. By default, the maximum size is 20 MB. Essbase allocates area in a dynamic calculator cache for data blocks until the maximum memory area specified by the DYNCALCACHEMAXSIZE setting is allocated. See "Changing the Dynamic Calculator Cache Size" on page 836.

Reviewing Dynamic Calculator Cache Usage

For each database, Essbase writes two messages to the application log for each data retrieval:

```
[Thu Oct 17 11:37:17 2007]Local/Sample///Info(1007125)
The number of Dynamic Calc Non-Store Members = [7 6 0 0 2 ]
```

```
[Thu Oct 17 11:37:17 2007]Local/Sample///Info(1007126)
The number of Dynamic Calc Store Members = [0 0 0 0 0 ]
```

The first message describes the total time required for the retrieval. If a dynamic calculator cache is used, the second message displays the number of blocks calculated within the data calculator cache (DCC = *n*) and the number of blocks calculated in general memory (non-DCC = *n*).

Changing the Dynamic Calculator Cache Size

Five configuration file settings are relevant to dynamic calculator caches. The optimum values for these dynamic calculator cache settings depend on the memory on the server machine, the configuration of all databases on the server machine, and the nature of user queries.

Table 109 describes each setting and includes recommendations for how to determine values for your system. To match your site's unique requirements, you may need to test and adjust the settings.

Table 109 essbase.cfg Settings for Dynamic Calculator Caches

DYNCALCCACHEMAXSIZE	
Description	This setting specifies the maximum size Essbase can allocate to each dynamic calculator cache on the server.
Recommended Setting	Recommended setting value = C * S * U. ● C is the value of the appropriate CALCLOCKBLOCK setting in the `essbase.cfg` file. (The SET LOCKBLOCK command specifies which CALCLOCKBLOCK setting to use.) ● S is the size of the largest expanded block across all databases on the machine. To calculate the expanded block size, multiply the number of members (including Dynamic Calc members and dynamic time series members) in each dense dimension together for the number of cells in the block, and multiply the number of cells by the size of each member cell, 8 bytes. For example, consider the member count in dense dimensions in Sample.Basic: - 19 (Year, with 12 stored members and 7 Dynamic Calc members, including HTD and QTD) - 14 (Measures, with 8 stored members and 6 Dynamic Calc members) - 4 (Scenario, with 2 stored members and 2 Dynamic Calc members)

DYNCALCCACHEMAXSIZE	
	Note: Label Only members are not counted. S = 19 * 14 * 4 cells (8 bytes / cell) = 8512 bytes per block This number is shown in the application log as the logical block size. ● U is the maximum number of expected concurrent users on the database that has the largest number of concurrent users. Assigning the value 0 (zero) to DYNCALCCACHEMAXSIZE tells Essbase not to use dynamic calculator caches. By default, the maximum size for this value is 20 MB (20,971,520 bytes).

DYNCALCCACHEWAITFORBLK	
Description	If Essbase uses all of the area allocated for a dynamic calculator cache, this setting tells Essbase whether to wait until space becomes available in the cache or to immediately write and calculate the blocks in memory outside the dynamic calculator cache. If the dynamic calculator cache is too small, it is possible for multiple threads to be in queue, each thread waiting to calculate its data blocks.
Recommended Setting	Recommended setting value = FALSE (default value). Before setting to TRUE, try these alternatives: ● Add physical memory to the server machine ● Increase the value of DYNCALCCACHEMAXSIZE, test, and repeat until you verify that you cannot use any more memory for the dynamic calculator cache.

DYNCALCCACHEBLKTIMEOUT	
Description	If Essbase is to wait for available space in the dynamic calculator cache, this setting defines how long it waits.
Recommended Setting	Recommended setting value = WT / B. ● WT is the maximum tolerable wait time for a query; for example, 5 seconds. ● B is the total number of logical blocks accessed in the largest query. To determine the value of B, check the messages in the application log for the largest number of Dyn.Calc.Cache "Big Block Allocs" for a query, as discussed in "Reviewing Dynamic Calculator Cache Usage" on page 430.

DYNCALCCACHEBLKRELEASE	
Description	If Essbase has waited the specified time and space still is not available in the dynamic calculator cache, this setting tells Essbase whether to write and calculate the blocks immediately outside the dynamic calculator cache or to create space in the dynamic calculator cache by swapping out blocks and temporarily compressing the swapped blocks in a dynamic calculator cache compressed-block buffer.
Recommended Setting	Recommended setting value = FALSE (default value). Set to TRUE only if you are experiencing severe memory shortage problems.

DYNCALCCACHECOMPRBLKBUFSIZE	
Description	If Essbase has waited the specified wait time and the DYNCALCCACHEBLKRELEASE setting is TRUE, this setting is the size of the dynamic calculator cache compressed-block buffer.
Recommended Setting	Recommended setting value = (C * S) / 2. ● C is the value of the current CALCLOCKBLOCK setting in the `essbase.cfg` file. The SET LOCKBLOCK command specifies which CALCLOCKBLOCK configuration setting is current.

DYNCALCCACHEMAXSIZE	
	● S is the size of the largest expanded block across all databases on the machine. Calculate S as described for the DYNCALCCACHEMAXSIZE setting.

Note:

After changing any parameter in the `essbase.cfg` file, you must stop and restart Essbase Server to use the new values.

For information about specific dynamic calculator cache settings, see the *Oracle Essbase Technical Reference*.

Fine-Tuning Cache Settings

After using a database at your site with typical data, user access, and standard environment (including servers, network, etc.), observe how Essbase performs. It is difficult to predict optimal cache sizes without testing. You may need to adjust your cache settings.

Understanding Cache Settings

The sizes of the index cache and the data file cache (when direct I/O is used) are the most critical Essbase cache settings. In general, the larger these caches, the less swapping activity occurs; however, setting larger cache sizes does not always improve performance. Read this entire section to understand cache size considerations.

Index Cache

The advantages of a large index cache plateau at some point. When the index cache size equals or exceeds the index size (including all index files on all volumes), performance does not improve. However, to account for future growth of the index, you can set the index cache size larger than the current index size. Because the index cache is filled with index pages, for optimum use of storage, set the size of the index cache to be a multiple of the size of the index page (8 KB). See "Index Files" on page 1028 for an example of estimating index size.

Data File Cache

If possible, set the data file cache to equal the size of the stored data, which is the combined size of all `ess*.pag` files. Otherwise, the data file cache should be as large as possible. If you want to account for future growth of stored data, you can set the data file cache size larger than the current size of stored data.

Note:

The data file cache is used only if you are using direct I/O.

Data Cache

The data cache should be about 0.125 times the data file cache. However, certain calculations require a larger data cache size. If many concurrent users are accessing different data blocks, this cache should be larger.

In general, if you must choose between allocating memory to the data file cache or allocating it to the data cache, choose the data file cache if you are using direct I/O. If upgrading from a previous version of Essbase, see the *Oracle Hyperion Enterprise Performance Management System Installation and Configuration Guide*.

Checking Cache Hit Ratios

Every cache has a “hit ratio”: the percentage of time that a requested piece of information is available in it. You can check the hit ratio of the index cache, the data cache, and the data file cache to determine whether to increase the cache size.

➤ To check cache hit ratios, see “Checking Cache Hit Ratios” in *Oracle Essbase Administration Services Online Help*.

- The cache hit ratio indicates the percentage of time that a requested piece of information is already in the cache. A higher hit ratio indicates that the data is in the cache more often. This improves performance, because the requested data need not be retrieved from disk for the next process. A hit ratio of 1.0 indicates that every time data is requested, it is found in the cache. This is the maximum performance possible from a cache setting.
- The Hit Ratio on Index Cache setting indicates the Essbase kernel success rate in locating index information in the index cache without having to retrieve another index page from disk.
- The Hit Ratio on Data File Cache setting indicates the Essbase kernel success rate in locating data file pages in the data file cache without having to retrieve the data file from disk.
- The Hit Ratio on Data Cache setting indicates the Essbase success rate in locating data blocks in the data cache without having to retrieve the block from the data file cache.
- Check memory allocation. Add smaller amounts of memory at a time , if needed, because a smaller increment may have the same benefit as a large one. Large, incremental allocations of memory usually result in very little gain in the hit ratio.

Checking Performance

You can check cache statistics for a database by using the **query database** MaxL statement with the **performance statistics** grammar. See Chapter 50, “Monitoring Performance.”

Running Test Calculations

Because calculations are the most processor-intensive operations on a Essbase database, you should run test calculations and examine how various cache sizes affect memory use on Essbase Server.

53 Optimizing Database Restructuring

In This Chapter

The information in this chapter applies only to block storage databases and is not relevant to aggregate storage databases.

Also see:

- Chapter 57, “Comparison of Aggregate and Block Storage”
- “Designing an Outline to Optimize Performance” on page 95
- “Positioning Dimensions and Members” on page 130
- “Using Alternative Design Approaches” on page 169
- “Optimizing Outline Performance” on page 170
- “Optimizing Calculation and Retrieval Performance” on page 177

Database Restructuring

As your business changes, you change the Essbase database outline to capture new product lines, provide information on new scenarios, reflect new time periods, and so on. Some changes to a database outline affect the data storage arrangement, forcing Essbase to restructure the database.

Because changes that require restructuring the database are time-consuming (unless you discard the data before restructuring), consider deciding on such changes based on how they affect performance. This section provides the information necessary to understand how restructuring affects performance and describes tasks you can perform related to database restructuring:

Note:

For information about clearing data and thus avoiding some restructuring, see CLEARDATA and CLEARBLOCK in the *Oracle Essbase Technical Reference* or Clearing Data in the *Oracle Essbase Administration Services Online Help*.

Types of Database Restructuring

This section describes the two ways that a database restructure is triggered.

Implicit Restructures

Essbase initiates an implicit restructure of the database files after an outline is changed using Outline Editor or Dimension Build. The type of restructure that is performed depends on the type of changes made to the outline:

- **Dense restructure:** If a member of a dense dimension is moved, deleted, or added, Essbase restructures the blocks in the data files and creates new data files. When Essbase restructures the data blocks, it regenerates the index automatically so that index entries point to the new data blocks. Empty blocks are not removed. Essbase marks all restructured blocks as dirty, so after a dense restructure you must recalculate the database. Dense restructuring, the most time-consuming of the restructures, can take a long time to complete for large databases.
- **Sparse restructure:** If a member of a sparse dimension is moved, deleted, or added, Essbase restructures the index and creates new index files. Restructuring the index is relatively fast; the time required depends on the index size.
- **Outline-only restructure:** If a change affects only the database outline, Essbase does not restructure the index or data files. Member name changes, creation of aliases, and dynamic calculation formula changes are examples of changes that affect only the database outline.

If you use incremental restructuring, Essbase defers dense restructuring. If you change a database outline frequently, consider enabling incremental restructuring. See "Incremental Restructuring and Performance" on page 845 for a comprehensive discussion of incremental restructuring.

Note:

How a database outline is changed (by using Outline Editor or using dimension build) does not influence restructuring. Only the type of information change influences what type of restructuring, if any, takes place. For information about outline changes and the type of restructures they cause, see "Outline Change Quick Reference" on page 847.

Explicit Restructures

When you manually initiate a database restructure, you perform an explicit restructure. An explicit restructure forces a full restructure of the database. A full restructure comprises a dense restructure plus removal of empty blocks.

➤ To initiate an full restructure, use a tool:

Tool	Topic	Location
Administration Services	Restructuring Databases Manually	*Oracle Essbase Administration Services Online Help*
MaxL	**alter database**	*Oracle Essbase Technical Reference*

Conditions Affecting Database Restructuring

Intelligent Calculation, name changes, and formula changes affect database restructuring:

- If you use Intelligent Calculation in the database, all restructured blocks are marked as dirty whenever data blocks are restructured. Marking the blocks as dirty forces the next default Intelligent Calculation to be a full calculation.
- If you change a name or a formula, Essbase does not mark the affected blocks as dirty. Therefore, you must use a method other than full calculation to recalculate the member or the database.

Use Table 110 for information about restructuring:

Table 110 Topics Related To Database Restructuring

Topic	Related Information
Intelligent Calculation	"Restructuring Databases" on page 416
Sparse and dense dimensions	● "Sparse and Dense Dimensions" on page 66, "Selection of Dense and Sparse Dimensions" on page 67 ● "Dense and Sparse Selection Scenarios" on page 69
Attribute dimensions	"Designing Attribute Dimensions" on page 168
Dimension building	Chapter 16, "Understanding Data Loading and Dimension Building"
Outline Editor	Chapter 7, "Creating and Changing Database Outlines"

Temporary Files Used During Restructuring

When Essbase restructures both the data blocks and the index, it uses the files described in Table 111:

Table 111 Files Used During Database Restructuring

File	Description
`ess`*`xxxxx`*`.pag`	Essbase data file
`ess`*`xxxxx`*`.ind`	Essbase index file
`dbname``.esm`	Essbase kernel file that contains control information used for database recovery
`dbname``.tct`	Transaction control table
`dbname``.ind`	Free fragment file for data and index free fragments
`dbname``.otl`	Outline file in which is defined all metadata for a database and how data is stored.

Dense Restructures

➤ To perform a dense restructure, Essbase does the following:

1 **Creates temporary files that are copies of the following files:**

   `ess`*`xxxxx`*`.ind`
   `ess`*`xxxxx`*`.pag`
   `dbname``.otl`
   `dbname``.esm`
   `dbname``.tct`

   Each temporary file substitutes either `n` or `u` for the last character of the file extension. Temporary file names are:

   `ess`*`xxxxx`*`.inn`
   `ess`*`xxxxx`*`.pan`
   `dbname``.otn`
   `dbname``.esn`
   `dbname``.tcu`

2 **Reads the blocks from the database files copied in step 1, restructures the blocks in memory, and stores them in the new temporary files. This step takes the most time.**

3 **Removes the database files that were copied in step 1**

4 **Renames the temporary files created in step 1 to the correct file names.**

Sparse Restructures

When Essbase does a sparse restructure (restructures only the index), it uses the following files:

- `ess`*`xxxxx`*`.ind`
- *`dbname`*`.otl`
- *`dbname`*`.esm`

➤ To perform a sparse restructure, Essbase does the following:

1 **Renames the** *`dbame`*`.esm` **file to** *`dbname`*`.esr`**.**

2 **Renames the** `ess`*`xxxxx`*`.ind` **files to** `ess`*`xxxxx`*`.inm`**.**

3 **Creates index files (**`ess`*`xxxxx`*`.ind`**) to store index information that is changed by the restructuring operation.**

4 **Removes** *`dbname`*`.esr` **and** `ess`*`xxxxx`*`.inm` **created in step 1.**

Optimization of Restructure Operations

If a database outline changes frequently, analyze the outline and the types of changes that you are making. Changes to sparse dimensions or attribute dimensions are relatively fast, because only the index changes. Changes to dense dimensions are relatively slow, because data blocks are rebuilt.

These types of restructure operations are listed from fastest to slowest:

- Outline only (no index or data files)
- Sparse (only index files)
- Dense (index files and data files) as a result of adding, deleting, or moving members and other operations. See “Outline Change Quick Reference” on page 847
- Dense (index and data files) as a result of changing a dense dimension to sparse or changing a sparse dimension to dense

Actions That Improve Performance

Several actions improve performance related to database restructuring:

- If you change a dimension frequently, make it sparse.
- Use incremental restructuring to control when Essbase performs a required database restructuring.
- Select options when you save a modified outline that reduce the amount of restructuring required.

Note:

Setting the isolation level to committed access may increase memory and time requirements for database restructure. Consider setting the isolation level to uncommitted access before a database restructure. For information on isolation level settings, see Chapter 48, “Ensuring Data Integrity.”

Incremental Restructuring and Performance

If you make frequent changes to a database outline, consider enabling incremental restructuring. With it enabled, Essbase defers restructuring so that a change to the database outline or to a dimension does not cause structural change. Essbase restructures the index and, if necessary, the affected block the next time the block is accessed.

Understanding Incremental Restructuring

When incremental restructuring is enabled, Essbase defers restructuring for the database changes listed in “Outline Change Quick Reference” on page 847, unless otherwise noted.

The following changes override incremental restructuring; that is, they result in immediate restructuring, even if incremental restructuring is enabled:

- Adding or deleting a nonattribute dimension.
- Deleting a stored member of a sparse dimension.
- Changing a dimension definition from sparse to dense or from dense to sparse.

- If you are using LROs in a database, incremental restructuring is automatically disabled on that database. Disabling incremental restructuring does not affect other databases on the server.
- Certain member additions and certain changes to sparse dimensions can also trigger immediate restructuring. See "Outline Change Quick Reference" on page 847.

Whether or not incremental restructuring is enabled, if an outline has already been incrementally restructured (a dense restructure is pending), adding shared members causes Essbase to perform a dense restructure.

Note:

Recalculate the database after any type of restructure operation.

Using Incremental Restructuring

You can enable incremental restructuring for any of the following databases:

- One database in an application
- All databases in an application
- All databases in all applications

To enable incremental restructuring, use the INCRESTRUC configuration setting in the `essbase.cfg` file. See the *Oracle Essbase Technical Reference*.

Essbase logs outline changes in an internal file, `*dbname*.ocl`. Essbase clears the file when it does a dense restructure or when you clear or reset the database. The file `*dbname*.ocl` can grow quite large. To clear this file, issue VALIDATE in ESSCMD, which causes Essbase to restructure any blocks whose restructure was deferred. When you issue VALIDATE, ensure that the database is *not* in read-only mode (which is used for backing up a database). See "Using VALIDATE to Check Integrity" on page 788.

Options for Saving a Modified Outline

Essbase displays a dialog box when you save outline changes that trigger database restructuring (using Outline Editor). In the Restructure Database dialog box, you define how data values are handled during restructure; for example, you can preserve all data, preserve only level 0 or input data, or discard all data during restructure. See "Saving Outlines" in *Oracle Essbase Administration Services Online Help*.

If the database contains data, you need enough free disk space on the server to create a backup copy of the database. Backup ensures that any abnormal termination during the restructure process does not corrupt the database.

Essbase may display a "Restructuring not required" message yet still perform an index-only restructure. This event most likely will occur if you make changes to a sparse dimension. If you try to cancel a restructure operation, Essbase may issue a "Can't cancel" message. If such a message is displayed, Essbase is performing final cleanup, and it is too late to cancel.

Outline Change Log

If you activate the outline change log, Essbase records all activity that affects the outline (member name changes, member moves, and so on). The more changes you make to the outline, the more updates Essbase must make to the log, slowing performance.

By default, Essbase does not log outline changes. To see whether outline logging is slowing performance, look for OUTLINECHANGELOG TRUE in the `essbase.cfg` file. See "Understanding and Using the Outline Change Log" on page 745.

Essbase Partitioning Option

When you use Partitioning, Essbase tracks outline changes so that you can synchronize the database outlines across partitions. Tracking outline changes slows restructuring, particularly when there are many structural changes.

If Essbase restructures data when you are using partitioning, perform the following steps to make sure that data is synchronized across partitions:

1. Validate the partitions.

   See "Validating Partitions" on page 246.

   Note:

   To validate a partition, you must have Database Manager permissions or higher.
2. Synchronize the outlines of the partitions.

   See "Synchronizing Outlines" on page 248.

Outline Change Quick Reference

The tables in this section show all outline changes that affect calculation and restructuring, including incremental restructuring.

Note:

If you are using Partitioning, restructuring affects only the database to which you are connected.

Table 112 Actions: Delete, Add, or Move Member

Action	Calculation and Standard Restructure Effects	Incremental Restructuring Applies? (If Enabled)
Delete member of sparse dimension	Data must be recalculated to reflect changes to relationships. Essbase deletes from the index file all pointers to blocks represented by the deleted member. Because the blocks are no longer	For regular members, no. Essbase restructures the index, overriding incremental restructure. For label-only members, yes, restructuring is deferred.

Action	Calculation and Standard Restructure Effects	Incremental Restructuring Applies? (If Enabled)
	pointed to, they become free space. No restructure.	
Delete member of attribute dimension	None	No
Delete member of dense dimension	Data must be recalculated to reflect changes to relationships. Essbase restructures the data files to reflect a changed block size. Essbase restructures the index.	Yes. Restructure deferred.
Delete shared member in sparse or dense dimension	Data must be recalculated. The data remains associated with the original member name, but, because the parent of the shared member may have depended on child data, recalculation is needed. No restructure.	No
Add member to sparse dimension	Data for the new member must be loaded or calculated to derive new values. Essbase restructures the index.	Yes. Restructure deferred.
Add member to dense dimension	Data for the new member must be loaded or calculated to derive new values. Data must be recalculated. Essbase restructures the data files to reflect a changed block size. Essbase restructures the index.	Yes. Restructure deferred.
Add member to attribute dimension	None	No
Add shared member to sparse or dense dimension	Data must be recalculated. The new shared member affects the consolidation to its parent. No restructure.	No
Move regular member within a sparse dimension	Data must be recalculated to reflect changes in consolidation. Essbase restructures the index file.	No. Essbase restructures the index file, overriding incremental restructure.
Move regular member within a dense dimension	Data must be recalculated to reflect changes in consolidation. Essbase restructures index and data files.	Yes. Restructure deferred.
Move an attribute dimension member	None	No

Table 113 Actions: Other Member-Related Changes

Action	Calculation and Standard Restructure Effects	Incremental Restructuring Applies? (If Enabled)
Change a member alias or add an alias to a member	None	No
Rename member	None	No
Change member formula	Data must be recalculated to reflect formula changes. No restructure.	No

Table 114 Actions: Dynamic Calculation-Related Changes

Action	Calculation and Standard Restructure Effects	Incremental Restructuring Applies? (If Enabled)
Define Dynamic Calc member as Dynamic Calc and Store	For dense dimension members: Essbase restructures both index and data files. For sparse dimension members: no restructure.	Yes. Restructure deferred.
Define Dynamic Calc and Store member as Dynamic Calc	None	No
Define regular dense dimension member as Dynamic Calc and Store	None	No
Define regular dense dimension member as Dynamic Calc	Essbase restructures both index and data files.	Restructure deferred.
Define sparse dimension Dynamic Calc and Store member or Dynamic Calc member as regular member	No restructure	No
Define sparse dimension regular member as Dynamic Calc or Dynamic Calc and Store	Essbase restructures index and data files.	Yes. Restructure deferred.
Define dense dimension Dynamic Calc and Store member as regular member	No restructure	No
Define dense dimension Dynamic Calc member as regular member	Essbase restructures index and data files.	Yes. Restructure deferred.
Define dense dimension regular member as Dynamic Calc member	Essbase restructures index and data files.	Yes. Restructure deferred.
Add, delete, or move sparse dimension Dynamic Calc member	Essbase restructures index files.	For member add or delete, restructure is deferred. For member move, Essbase restructures index files, overriding incremental restructure.
Add, delete, or move sparse dimension Dynamic Calc and Store member	Essbase restructures index files.	For member add, restructure deferred.

Action	Calculation and Standard Restructure Effects	Incremental Restructuring Applies? (If Enabled)
		For member move or delete, Essbase restructures index files (overrides incremental restructure).
Add, delete, or move dense dimension Dynamic Calc and Store member	Essbase restructures index and data files.	No
Add, delete, or move dense dimension Dynamic Calc member	No restructure.	No

Table 115 Actions: Property and Other Changes

Action	Calculation and Standard Restructure Effects	Incremental Restructuring Applies? (If Enabled)
Change dense-sparse property	Data must be recalculated. Essbase restructures both index and data files.	Essbase restructures index and data files overriding incremental restructure.
Change label only property	Data must be recalculated. Essbase restructures index and data files.	Restructure deferred.
Change shared member property	Data must be recalculated to reflect the changed data value of the child. Essbase restructures both index and data files.	Restructure deferred.
Change properties other than dense-sparse, label, or shared	Data may need to be recalculated to reflect changed consolidation properties, such as changing time balance from first to last.	No
Change the order of two sparse dimensions	No calculation or data load impact. Essbase restructures the index.	Essbase restructures the index, overriding incremental restructure.
Change the order of dimensions	Data must be recalculated. Essbase restructures both index and data files.	Essbase restructures index and data files (overrides incremental restructure).
Change the order of attribute dimensions	None	No
Create, delete, clear, rename, or copy an alias table	None	No
Import an alias table or set a member alias	None	No
Change the case-sensitive setting	None	No
Name a level and generation	None	No
Create, change, or delete a UDA	None	No

54

Optimizing Data Loads

In This Chapter

Some information in this chapter applies only to block storage databases and is not relevant to aggregate storage databases. Also see Chapter 57, "Comparison of Aggregate and Block Storage."

Understanding Data Loads

This section does not apply to aggregate storage databases.

Loading a large data source into an Essbase database can take hours. You can shorten the data loading process by minimizing the time spent on these actions:

- Reading and parsing the data source
- Reading and writing to the database

To optimize data load performance, think in terms of database structure. Essbase loads data block by block. For each unique combination of sparse dimension members, one data block contains the data for all the dense dimension combinations, assuming that at least one cell contains data. For faster access to block locations, Essbase uses an *index*. Each entry in the index corresponds to one data block. See "Sparse and Dense Dimensions" on page 66, "Selection of Dense and Sparse Dimensions" on page 67, and "Dense and Sparse Selection Scenarios" on page 69.

When Essbase loads a data source, Essbase processes the data in three main stages:

- **Input:** Essbase reads a portion of the data source.
- **Preparation:** Essbase arranges the data in preparation for putting it into blocks.

- **Write:** Essbase puts the data into blocks in memory and then writes the blocks to disk, finding the correct block on the disk by using the index, which is composed of pointers based on sparse intersections.

This process is repeated until all data is loaded. By using one or more processing threads in each stage, Essbase can perform some processes in parallel. See "Managing Parallel Data Load Processing" on page 855.

Examples in this chapter assume that you are familiar with the following topic: "Data Sources" on page 258.

Grouping Sparse Member Combinations

This section does not apply to aggregate storage databases.

The most effective strategy to improve performance is to minimize the number of disk I/Os that Essbase must perform while reading or writing to the database. Because Essbase loads data block by block, organizing the source data to correspond to the physical block organization reduces the number of physical disk I/Os that Essbase must perform.

Arrange the data source so that records with the same unique combination of sparse dimensions are grouped together. This arrangement corresponds to blocks in the database.

The examples in this chapter illustrate ways that you can organize the data following this strategy. These examples use a subset of the Sample.Basic database, as shown in Table 116:

Table 116 Dimensions and Values for Examples

Sparse, Nonattribute Dimensions	Dense Dimensions
Scenario (Budget, Actual)	Measures (Sales, Margin, COG, Profit)
Product (Cola, Root Beer)	Year (Jan, Feb)
Market (Florida, Ohio)	

Note:

Because you do not load data into attribute dimensions, they are not relevant to this discussion although they are sparse.

Consider the following data source. Because it is not grouped by sparse-dimension member combinations, this data has not been sorted for optimization. As Essbase reads each record, it must deal with different members of the sparse dimensions.

```
Jan
Actual    Cola          Ohio      Sales    25
Budget    "Root Beer"   Florida   Sales    28
Actual    "Root Beer"   Ohio      Sales    18
Budget    Cola          Florida   Sales    30
```

This data loads slowly because Essbase accesses four blocks instead of one.

An optimally organized data source for the same Sample.Basic database shows different records sorted by a unique combination of sparse-dimension members: Actual -> Cola -> Ohio. Essbase accesses only one block to load these records.

```
Actual     Cola    Ohio    Jan   Sales     25
Actual     Cola    Ohio    Jan   Margin    18
Actual     Cola    Ohio    Jan   COGS      20
Actual     Cola    Ohio    Jan   Profit     5
```

You can use a data source that loads many cells per record. Ensure that records are grouped together by unique sparse-dimension member combinations. Then order the records so that the dimension in the record for which you provide multiple values is a dense dimension.

The next data source example uses a header record to identify the members of the Measures dimension, which is dense. The data is sorted first by members of the dense dimension Year and grouped hierarchically by members of the other dimensions. Multiple values for the Measures dimension are provided on each record.

```
                                 Sales  Margin   COG  Profit
Jan Actual  Cola         Ohio       25      18    20       5
Jan Actual  Cola         Florida    30      19    20      10
Jan Actual  "Root Beer"  Ohio       18      12    10       8
Jan Actual  "Root Beer"  Florida    28      18    20       8
```

Notice that the heading and first data line that requires two lines in this example; the previous example needs four lines for the same data.

For information about arranging data in source files before loading, see "Data Sources that Do Not Need a Rules File" on page 264.

Making the Data Source as Small as Possible

Make the data source as small as possible. The fewer fields that Essbase reads in the data source, the less time is needed to read and load the data.

Group the data into ranges. Eliminating redundancy in the data source reduces the number of fields that Essbase must read before loading data values.

The following example data source is not organized in ranges. It includes unneeded repetition of fields. All values are Profit values. Profit must be included only at the beginning of the group of data applicable to it. This example contains 33 fields that Essbase must read to load the data values properly.

```
Profit
Jan     "New York"   Cola          4
Jan     "New York"   "Diet Cola"   3
Jan     Ohio         Cola          8
Jan     Ohio         "Diet Cola"   7
Feb     "New York"   Cola          6
Feb     "New York"   "Diet Cola"   8
Feb     Ohio         Cola          7
Feb     Ohio         "Diet Cola"   9
```

The next example provides the same data optimized by grouping members in ranges. By eliminating redundancy, this example contains only 23 fields that Essbase must read in order to load the data values properly.

```
Profit
Jan    "New York"    Cola           4
                     "Diet Cola"    3
       Ohio          Cola           8
                     "Diet Cola"    7
Feb    "New York"    Cola           6
                     "Diet Cola"    8
       Ohio          Cola           7
                     "Diet Cola"    9
```

Essbase assigns the first value, 4, to Jan->New York->Cola; it assigns the next value, 3, to Jan->New York->Diet Cola and so on.

Although sorted efficiently, the data source sorted and grouped by dense dimensions shows a lot of repetition that can slow down the load process. You can further optimize this data by grouping the data into ranges. The optimized data source below eliminates the redundant fields, reducing processing time.

```
                                Sales  Margin   COG  Profit
Jan Actual   Cola         Ohio     25      18    20       5
                          Florida  30      19    20      10
             "Root Beer"  Ohio     18      12    10       8
                          Florida  28      18    20       8
```

See "Formatting Ranges of Member Fields" on page 265.

Making Source Fields as Small as Possible

Making fields in a data source smaller enables Essbase to read and load faster.

Make the fields in the data source as small as possible by performing the following tasks:

- Remove excess white space in the data source. For example, use tabs instead of blank spaces.
- Round computer-generated numbers to the precision you need. For example, if the data value has nine decimal points and you care about two, round the number to two decimal points.
- Use #MI instead of #MISSING.

Positioning Data in the Same Order as the Outline

This section does not apply to aggregate storage databases.

The index is organized in the same order as the sparse dimensions in the outline. To further optimize the data source, with the sparse data combinations in the data source grouped together, arrange the data so that sparse dimensions are in the same order as the outline.

Essbase pages portions of the index in and out of memory as requested by the data load or other operations. Arranging the source data to match the order of entries in the index speeds the data load because it requires less paging of the index. Less paging results in fewer I/O operations.

Essbase uses the index cache size to determine how much of the index can be paged into memory. Adjusting the size of the index cache may also improve data load performance.

Note:

If the index cache size is large enough to hold the entire index in memory, positioning data in the same order as the outline does not affect the speed of data loads.

See "Sizing the Index Cache" on page 827.

Loading from Essbase Server

Loading the data source from Essbase Server is faster than loading from a client computer. To load a data source from the server, move the data source to the server and start the load.

Loading data from the server improves performance because the data need not be transported over the network from the client computer to the server computer.

Managing Parallel Data Load Processing

The methods described earlier in this chapter give you the most substantial data load performance enhancements. If you have not done so, carefully evaluate your processor speed and memory requirements and upgrade your computers to meet them.

Another way to speed data loads is to work with the Essbase parallel data load feature to optimize processor resources. The parallel data load feature recognizes opportunities to process data load tasks at the same time. Although some opportunities present themselves on single-processor computers, many more opportunities are available on multiple-processor computers.

To fine-tune processor use for specific application and database situations, Essbase provides these `essbase.cfg` settings: DLTHREADSPREPARE, DLTHREADSWRITE, and DLSINGLETHREADPERSTAGE.

Understanding Parallel Data Load Processing

When Essbase loads a data source, it works with a portion of data at a time. Essbase looks at each stage as a task and uses separate processing *threads* in memory to perform each task.

One form of parallel processing occurs when one thread takes advantage of processor resources that are left idle during the wait time of another thread. For example, while a thread performs I/O processing, it must wait for the slower hardware to perform its task. While this thread waits, another thread can use the idle processor resource. Processing staged tasks in parallel can improve processor efficiency by minimizing idle time.

When computers have multiple processors, Essbase can perform an additional form of parallel processing. When a data load stage completes its work on a portion of data, it can pass the work to the next stage and start work immediately on another portion of data. Processing threads perform their tasks simultaneously on the different processors, providing even faster throughput.

Optimizing Parallel Data Load Processing

Although Essbase uses parallel processing to optimize processor resources across the data load stages, processor resources are idle at times. To take advantage of these times, Essbase can further divide record processing in the preparation and write stages. To tailor parallel processing to your situation, you can use the DLTHREADSPREPARE and DLTHREADSWRITE `essbase.cfg` settings to tell Essbase to use additional threads during these stages.

Setting Parallel Data Load Settings

As shown in Table 117, Essbase provides three `essbase.cfg` settings that enable you to manage parallel data load processing.

You can specify setting values that apply to all applications on a given Essbase Server or specify settings multiple times with different values for different applications and databases.

Table 117 Parallel Data Load essbase.cfg Settings

Setting	Description
DLTHREADSPREPARE	Specifies how many threads Essbase may use during the data load stage that codifies and organizes the data in preparation to being written to blocks in memory.
DLTHREADSWRITE	Specifies how many threads Essbase may use during the data load stage that writes data to the disk. High values may require allocation of additional cache. See "Implications in Sizing the Data Cache" on page 856. **Note:** For aggregate storage databases, Essbase Server uses one thread with aggregate storage cache. The DLTHREADSWRITE setting is ignored.
DLSINGLETHREADPERSTAGE	Specifies that Essbase use a single thread per stage, ignoring the values in the DLTHREADSPREPARE and DLTHREADSWRITE settings.

Only when the DLSINGLETHREADPERSTAGE setting is set to FALSE for the specific application and database being loaded does the data load process use the thread values specified in the DLTHREADSPREPARE and DLTHREADSWRITE settings.

See the *Oracle Essbase Technical Reference*.

Implications in Sizing the Data Cache

For block storage databases, Essbase Server allocates the data cache memory area to hold uncompressed data blocks. Each thread specified by the DLTHREADSWRITE setting uses an area in the data cache equal to the size of an expanded block.

Depending on the size of the block, the number of threads, and how much data cache is used by other concurrent operations during a data load, it may be possible to need more data cache than is available. In such circumstances, decrease the number of threads or increase the size of the data cache.

See "Changing the Data Cache Size" on page 830.

Testing Different Thread Values

While processing data loads, you can view processor activity. Different operating systems provide different tools for viewing processor activity. For example, the Task Manager in Windows NT and Windows 2000 enables you to view processor and memory usage and processes. Among the tools available on UNIX are `top` and `vmstat`. You can also use third-party tools to view and analyze system use.

➤ To assess system use during data load processing;

1 **Start with the default parallel data load processing thread, in which Essbase uses a single thread per stage.**

2 **Perform and time the data load.**

3 **Monitor the entire process, identifying the stages during which the processor may be idle.**

4 **Alter the `essbase.cfg` settings described in "Setting Parallel Data Load Settings" on page 856.**

5 **Repeat the last three steps until you find values that provide the best performance.**

55

Optimizing Calculations

In This Chapter

The information in this chapter applies only to block storage databases and is not relevant to aggregate storage databases.

Also see:

- Chapter 57, "Comparison of Aggregate and Block Storage"
- Chapter 14, "Designing Partitioned Applications"
- Chapter 26, "Dynamically Calculating Data Values"
- Chapter 28, "Developing Calculation Scripts"
- Chapter 25, "Understanding Intelligent Calculation"

Designing for Calculation Performance

You can configure a database to optimize calculation performance.

The best configuration for the site depends on the nature and size of the database. Use the information in the following topics as guidelines only.

Block Size and Block Density

A data block size of 8 Kb to 100 Kb provides optimal performance in most cases.

If data blocks are much smaller than 8 KB, the index is usually very large, forcing Essbase to write to and retrieve the index from disk. This process slows calculation.

If data blocks are much larger than 100 KB, Intelligent Calculation does not work effectively. See Chapter 25, "Understanding Intelligent Calculation."

To optimize calculation performance and data storage, balance data block density and data block size by rearranging the dense and sparse dimension configuration of the database. Keep these suggestions in mind:

- Keep data block size between 8 KB and 100 KB with as high a block density as possible.
- Run test calculations of the most promising configurations of a database that contains representative data. Check results to determine the configuration that produces the best calculation performance.

You can view information about a database, including the potential and actual number of data blocks and the data block size.

➤ To view data block information, use a tool:

Tool	Topic	Location
Administration Services	Checking Data Block Statistics	*Oracle Essbase Administration Services Online Help*
ESSCMD	GETDBINFO	*Oracle Essbase Technical Reference*

Order of Sparse Dimensions

You may improve calculation performance by changing the order of standard (not attribute) sparse dimensions in the database outline. Order standard sparse dimensions by the number of members they contain, placing the dimension that contains the fewest members first. This arrangement provides many possible improvements, depending on the site:

- The calculator cache functions more effectively, providing approximately a 10% performance improvement if you have a database outline with a large dimension (for example, one containing 1000 members).
- Parallel calculation, if enabled, more likely will be used if the standard sparse dimension with the most members is the last standard sparse dimension in the outline.

Incremental Data Loading

Many companies load data incrementally. For example, a company may load data each month for that month.

To optimize calculation performance when you load data incrementally, make the dimension tagged as time a sparse dimension. If the time dimension is sparse, the database contains a data block for each time period. When you load data by time period, Essbase accesses fewer data blocks because fewer blocks contain the relevant time period. Thus, if you have Intelligent Calculation enabled, only the data blocks marked as dirty are recalculated. For example, if you load data for March, only the data blocks for March and the dependent parents of March are updated.

However, making the time dimension sparse when it is naturally dense may significantly increase the size of the index, creating possibly slower performance due to more physical I/O activity to accommodate the large index.

If the dimension tagged as time is dense, you still receive some benefit from Intelligent Calculation when you do a partial data load for a sparse dimension. For example, if Product is sparse and you load data for one product, Essbase recalculates only the blocks affected by the partial load, although time is dense and Intelligent Calculation is enabled.

For information on incremental loads, see "Loading Data into Aggregate Storage Databases" on page 957.

Database Outlines with Multiple Flat Dimensions

Calculation performance may be affected if a database outline has multiple flat dimensions. A flat dimension has very few parents, and each parent has many thousands of children; in other words, flat dimensions have many members and few levels.

You can improve performance for outlines with multiple flat dimensions by adding intermediate levels to the database outline.

Formulas and Calculation Scripts

You may achieve significant improvements in calculation performance by carefully grouping formulas and dimensions in a calculation script. In this way, you can ensure that Essbase cycles through the data blocks in the database as few times as possible during a calculation.

Order commands in calculation scripts to make the database calculation as simple as possible. Consider applying all formulas to the database outline and using a default calculation (CALC ALL). This method may improve calculation performance.

See Chapter 28, "Developing Calculation Scripts" and "Calculation Passes" on page 399.

Monitoring and Tracing Calculations

You can display information in the application log about how Essbase is calculating the database by using the following commands in a calculation script.

SET MSG SUMMARY and SET MSG DETAIL

You can use the SET MSG SUMMARY and SET MSG DETAIL calculation commands in a calculation script to do the following:

- Display calculation settings, for example, whether completion notice messages are enabled
- Provide statistics on the number of data blocks created, read, and written
- Provide statistics on the number of data cells calculated

SET MSG DETAIL also provides an information message every time Essbase calculates a data block. SET MSG DETAIL is useful for reviewing the calculation order of data blocks and for testing intelligent recalculations.

Caution!

Because SET MSG DETAIL causes a high processing overhead, use it only during test calculations.

SET MSG SUMMARY causes a processing overhead of approximately 1% to 5%, depending on database size, and is therefore appropriate for all calculations.

SET NOTICE

You can use the SET NOTICE calculation command in a calculation script to display calculation completion notices that tell you what percentage of the database has been calculated. You can use the SET MSG SUMMARY command with the SET NOTICE command to show calculation progress between completion notices. Completion notices do not significantly reduce calculation performance, except when used with a very small database.

Using Simulated Calculations to Estimate Calculation Time

You can simulate a calculation using SET MSG ONLY in a calculation script. A simulated calculation produces results that help you analyze the performance of a real calculation that is based on the same data and outline.

By running a simulated calculation with a command such as SET NOTICE HIGH, you can mark the relative amount of time each sparse dimension takes to complete. Then, by performing a real calculation on one or more dimensions, you can estimate how long the full calculation will take, because the time a simulated calculation takes to run is proportional to the time that the actual calculation takes to run.

For example, if the calculation starts at 9:50:00 AM, and the first notice is time-stamped at 09:50:10 AM and the second is time-stamped at 09:50:20 AM, you know that each of part of the calculation took 10 seconds. If you then run a real calculation on only the first portion and note that it took 30 seconds to run, you know that the other portion also will take 30 seconds. If there were two messages total, then you would know that the real calculation will take approximately 60 seconds (20 / 10 * 30 = 60 seconds).

Use the following topics to learn how to perform a simulated calculation and how to use a simulated calculation to estimate calculation time.

Performing a Simulated Calculation

Before you can estimate calculation time, you must perform a simulated calculation on a data model that is based on your actual database.

➤ To perform a simulated calculation:

1 **Create a data model that uses all dimensions and all levels of detail about which you want information.**

2 **Load all data. This procedure calculates only data loaded in the database.**

3 **Create a calculation script with these entries:**

```
SET MSG ONLY;

SET NOTICE HIGH;

CALC ALL;
```

If you are using dynamic calculations on dense dimensions, substitute the CALC ALL command with the specific dimensions that you need to calculate; for example, CALC DIM EAST.

Note:

If you try to validate the script, Essbase reports an error. Disregard the error.

4 **Run the script.**

5 **Find the first sparse calculation message in the application log and note the time in the message.**

6 **Note the time for each subsequent message.**

7 **Calculate the dense dimensions of the model that are not being dynamically calculated:**

```
CALC DIM (DENSE_DIM1, DENSE_DIM2, …);
```

8 **Calculate the sparse dimensions of the model:**

```
CALC DIM (SPARSEDIM1, SPARSEDIM2, …);
```

9 **Project the intervals at which notices will occur, and then verify against sparse calculation results. You can then estimate calculation time.**

Estimating Calculation Time

After you perform a simulated calculation, you record the results and use them to estimate actual calculation time.

➤ To estimate total calculation time:

1 **Note the times of all the intervals between application log messages generated by SET NOTICE HIGH.**

See Table 118.

2 **Use the following calculation to estimate the time for a real calculation:**

Total time required for simulated calculation, divided by the first simulated calculation notice interval, multiplied by the first real calculation time interval.

Table 118 Sample Intervals Between Log Messages

Calculation Notice Number	Simulated Calculation Time Interval (in seconds)	Sparse dimension Calculation Interval (in seconds)
1	7	45
2	5	
3	6	
4	3	
5	4	
6	2	
7	6	
8	4	
9	3	
10	3	
Total calculation time	43	

In this example, 43 / 7 * 45 = 276.4 seconds, so the real calculation should take 276.4 seconds.

Factors Affecting Estimate Accuracy

The simulated calculation should return a time accurate to about 5%, excluding the following issues:

- "Variations Due to a Chain of Influences" on page 864
- "Variations Due to Outline Structure" on page 865

When these factors are present, this estimating technique more closely predicts calculation time when Essbase reaches 30%–40% of the simulated calculations (30%–40% of the messages generated by SET NOTICE HIGH).See the *Oracle Essbase Technical Reference*.

Variations Due to a Chain of Influences

Using SET MSG ONLY as a calculation-time estimating technique should be validated against later CALCNOTICE intervals. The results of this estimating technique vary because of the following chain of influences:

1. Blocks differ in block density through the real consolidation process, therefore
2. The rate at which Essbase writes blocks to the disk differs, therefore
3. The rate at which blocks are processed in the cache differs, therefore
4. Actual results may differ from the predicted calculation time.

Variations Due to Outline Structure

Another factor that can make actual results diverge significantly from predicted is the outline structure. Calculations based on CALCNOTICE intervals assume evenly balanced processing time throughout the outline. Factors that can skew this balance include the following situations:

- The model contains one or two sparse dimensions that are large in relation to the other sparse dimensions.
- Larger dimensions have member configurations that result in multiple shared rollups.

Changing the Outline Based on Results

After you have estimated and analyzed a simulated calculation, you can make changes in the outline to improve performance.

From top to bottom in the outline, order sparse dimensions to create the fewest percentage increases in upper blocks:

- Level 0 blocks following full model load 100,000
- Upper level blocks after consolidating only sparse dimension 1: 1,000,000
- Upper level blocks after consolidating only sparse dimension 2: 3,000,000
- Upper level blocks after consolidating only sparse dimension 3: 10,000,000
- Upper level blocks after consolidating only sparse dimension 4: 300,000
- Upper level blocks after consolidating only sparse dimension 5: 5,700,000

For example:

- #4 (members = 10,000, 4 levels)
- #1 (members = 500, 2 levels)
- #2 (members = 100, 4 levels)
- #5 (members = 10,000, 4 levels)
- #3 (members = 20, flat)

Use the simulated calculation to generate the upper block count. These numbers may be accurate despite actual dimension sizes as noted next to the items above.

Caution!

The largest count of members is not always a good predictor.

Estimating Calculation Affects on Database Size

Given the current number of blocks in a database, you can estimate the number of blocks that a CALC ALL will produce.

➤ To estimate the database size resulting from a calculation using interactive mode:

1. **Load data and issue a CALC ALL command and note the average block size.**
2. **Start the MaxL shell, log into Essbase, and start an application and database.**

   ```
   essmsh
   login username password;
   alter system load application appname;
   alter application appname load database dbname;
   ```

3. **Providing the application and database name, enter the following MaxL statement and note the value that is returned for the number of blocks.**

   ```
   query database appname.dbname get estimated size;
   ```

4. **Multiply the number of blocks by the average size of the blocks in the database.**

   Results are accurate to ±10%.

   Be aware of the following conditions when you query Essbase for an estimate of the full size of a database:

   - You must perform this query after a CALC ALL. Any other calculation will not produce accurate results.
   - You can obtain accurate results with formulas only if they are on sparse dimensions.
   - You cannot obtain accurate results with top-down calculations on any member in combination with a lock on data (committed access).
   - If you need to estimate partitions, you must query Essbase for a database size estimate on every partition and add the results. If you query for the size of only the source database, the estimate includes only the data on the source database server.

Using Parallel Calculation

The following topics discuss parallel calculation and how it might improve performance for your site.

Parallel Calculation

Essbase provides two ways of invoking a calculation:

- The calculation may be implicitly specified by the outline itself.
- The calculation may be explicitly specified by a calculation script that you create. The script contains formulas and calculation instructions.

Regardless of how a calculation is triggered, Essbase can execute the calculation in one of two modes:

- *Serial calculation* is the default. With serial calculation, each calculation pass is scheduled to run on a single processor. If invoked from a calculation script, the calculations are executed sequentially in the order in which they appear in the calculation script.

- *Parallel calculation* breaks each calculation pass into sub-tasks. The sub-tasks that can run independently of one another are scheduled to run simultaneously on up to four threads. Each thread may be on a different processor.

➤ To change from the default serial calculation to parallel calculation, change, at most, two configuration settings and restart the server, or add an instruction to the calculation script.

See "Enabling Parallel Calculation" on page 871.

The following topics discuss the details of parallel calculation.

Essbase Analysis of Feasibility

Essbase evaluates whether using parallel calculation is possible before each calculation pass for which you have enabled parallel calculation.

Essbase analyzes the outline and the calculation requested for each calculation pass. Remember that one calculation may require multiple passes. Some situations may create the need for multiple passes, including dynamic calculation, the presence of a member tagged as two-pass, or calculations that create certain kinds of interdependencies. See "Calculation Passes" on page 399.

If Essbase determines that parallel calculation is possible, Essbase splits the calculation into smaller tasks that are independent of each other. During the calculation, Essbase performs the smaller tasks simultaneously.

However, Essbase uses serial calculation even if parallel calculation is enabled if there are complex interdependencies between formulas that participate in the pass. Such interdependencies render parallel calculation impossible.

Parallel Calculation Guidelines

Outline structure and application design determine whether enabling parallel calculation can improve calculation performance. Before you enable parallel calculation, review the following guidelines, which will help you get the full benefit of parallel calculation:

- Use the uncommitted access isolation level. Parallel calculation is not supported if you use the committed access isolation level. See "Uncommitted Access" on page 783.
- One or more formulas present in a calculation may prevent Essbase from using parallel calculation even if it is enabled. For a description of formulas that may force serial calculation regardless of parallel calculation settings, see "Formula Limitations" on page 868.
- Calculation tasks are usually generated along the last sparse dimension of an outline. Order the sparse dimensions in an outline from smallest to largest, based on actual size of the dimension as reported by the ESSCMD command GETDBSTATS. This ordering recommendation is consistent with recommendations for optimizing calculator cache size and consistent with other outline recommendations. For a description of situations that may need to use additional dimensions (more than the last sparse dimension) and for instructions on how to increase the number of sparse dimensions used, see "Identifying Additional Tasks for Parallel Calculation" on page 872.

- Parallel calculation is effective on nonpartitioned applications and these partitioned applications:
    - Replicated partitions
    - Linked partitions
    - Transparent partitions if the calculation occurs at the target database. The number of sparse dimensions specified by CALCTASKDIMS in the `essbase.cfg` file or by SET CALCTASKDIMS in a calculation script must be set at 1 (the default value). For information on limitations imposed by the use of parallel calculation with transparent partitions, see "Transparent Partition Limitations" on page 869; for information on using CALCTASKDIMS or SET CALCTASKDIMS, see "Identifying Additional Tasks for Parallel Calculation" on page 872.
- If you have selected incremental restructuring for a database and have made outline changes that are pending a restructure, do not use parallel calculation. Unpredictable results may occur.

Relationship Between Parallel Calculation and Other Essbase Features

The following topics discuss the relationship between parallel calculation and other Essbase functionality.

Retrieval Performance

Placing the largest sparse dimension at the end of the outline for maximum parallel calculation performance may slow retrieval performance. See "Optimizing Query Performance" on page 95.

Formula Limitations

The presence of some formulas may force serial calculation. The following formula placements likely will force serial calculation:

- A formula on a dense member, including all stored members and any Dynamic Calc members upon which a stored member may be dependent, that causes a dependence on a member of the dimension that is used to identify tasks for parallel calculation.
- A formula that contains references to variables declared in a calculation script that uses @VAR, @ARRAY, or @XREF.
- A sparse dimension member formula using @XREF, and the dimension for the sparse member is fully calculated. @XREF does not force serial calculation when it is on dense Dynamic Calc members that are not dependent on other stored members during the batch calculation.
- A member formula that causes a circular dependence. For example, member A has a formula referring to member B, and member B has a formula referring to member C, and member C has a formula referring to member A.

- A formula on a dense or sparse member with a dependency on a member or members from the dimension used to identify tasks for parallel processing.
- A sparse dimension member formula that contains references to members from other sparse dimensions.

If you need to use a formula that might prevent parallel calculation, you can either mark the member of the formula as Dynamic Calc or exclude it from the scope of the calculation. To see whether a formula is preventing parallel calculation, check the application log. For relevant error messages, see "Monitoring Parallel Calculation" on page 873.

Calculator Cache

At the start of a calculation pass, Essbase checks the calculator cache size and the degree of parallelism and then uses the calculator cache bitmap option appropriate for maximum performance. Therefore, the bitmap option used for parallel calculation may be different from that used for serial calculation.

For example, assume Essbase performs a serial calculation and uses multiple bitmaps and a single anchoring dimension. Without explicit change of the calculator cache size, Essbase might perform a parallel calculation using only a single bitmap and a single anchoring dimension.

You can determine the calculator cache mode that controls the bitmap options by checking the application log at the start of each calculation pass for an entry similar to the following:

```
Multiple bitmap mode calculator cache memory usage has a limit of [50000]
bitmaps.
```

When using parallel calculation in multiple bitmap mode, you may encounter high memory usage. If so, you can use the configuration setting PARCALCMULTIPLEBITMAPMEMOPT to optimize memory use in multiple bitmap mode. This setting can be used with, or separately from, MULTIPLEBITMAPMEMCHECK. To enable PARCALCMULTIPLEBITMAPMEMOPT, add the following line to your `essbase.cfg` file:

```
PARCALCMULTIPLEBITMAPMEMOPT TRUE
```

See "Sizing the Calculator Cache" on page 830.

Transparent Partition Limitations

Parallel calculation with transparent partitions has the following limitations:

- You cannot use parallel calculation across transparent partitions unless the calculation occurs at the target.
- You must set CALCTASKDIMS or SET CALCTASKDIMS to 1 (the default) so that there is only one anchoring dimension.
- You must increase the calculator cache so that multiple bitmaps can be used. You can identify the calculator cache mode that controls the bitmap options by checking the application log at the start of each calculation pass for an entry similar to the following:

  ```
  Multiple bitmap mode calculator cache memory usage has a limit of
  [50000] bitmaps.
  ```

See "Sizing the Calculator Cache" on page 830.

Restructuring Limitation

Do not use parallel calculation if you have selected incremental restructuring. Parallel calculation does not support incremental restructuring.

Commit Threshold Adjustments

Essbase checks the commit threshold specified by the database setting "Number of blocks before internal commit." If the setting requires that less than 10 MB of data be written before an internal commit, then Essbase automatically increases the commit threshold for the duration of the calculation pass to 10 MB. If the setting is greater than 10 MB, Essbase uses the setting value.

Essbase writes a message to the application log noting the temporary increase if it occurs.

If you can allocate more than 10 MB extra disk space for calculation, consider increasing the commit threshold value; that is, the number of blocks before a commit, to a very large number for better performance.

➤ To view the current threshold, use a tool:

Tool	Topic	Location
Administration Services	Setting Data Integrity Options	*Oracle Essbase Administration Services Online Help*
MaxL	**display database*dbs_name***	*Oracle Essbase Technical Reference*
ESSCMD	GETDBINFO: Number of blocks modified before internal commit	*Oracle Essbase Technical Reference*

➤ To modify the commit threshold, use a tool:

Tool	Topic	Location
Administration Services	Setting Data Integrity Options	*Oracle Essbase Administration Services Online Help*
MaxL	**alter database *dbs_name* set implicit_commit after *n*blocks**	*Oracle Essbase Technical Reference*, list of MaxL statements
ESSCMD	SETDBSTATEITEM 21	"Example of Specifying Isolation Level Settings with ESSCMD" on page 787

See "Uncommitted Access" on page 783.

Isolation Level Limitation

You must use uncommitted mode for parallel calculation.

➤ To set the isolation level to uncommitted mode, use a tool:

Tool	Topic	Location
Administration Services	Setting Data Integrity Options	*Oracle Essbase Administration Services Online Help*
MaxL	**alter database dbs_name disable committed_mode**	*Oracle Essbase Technical Reference*, list of MaxL statements
ESSCMD	SETDBSTATEITEM 18	"Example of Specifying Isolation Level Settings with ESSCMD" on page 787

See "Uncommitted Access" on page 783.

Checking Current Parallel Calculation Settings

You can check either the server configuration file or the calculation script that you plan to use to see if parallel calculation is enabled.

➤ To check whether parallel calculation has been enabled in the server configuration file:

1 **Open the server `essbase.cfg` file with a text editor.**

2 **Search for the parameter CALCPARALLEL, and check its specified value.**

The number of threads that can simultaneously perform tasks to complete a calculation is specified by the value 1–4. See the *Oracle Essbase Technical Reference*.

➤ To check whether a calculation script sets parallel calculation, look for the SET CALCPARALLEL command. Review the script carefully, because the script may enable or disable parallel calculation more than once.

Enabling Parallel Calculation

To use parallel calculation, enable it at the server level, application level, or database level using either of these methods:

- Add or edit the appropriate configuration settings to the `essbase.cfg` file.

  See CALCPARALLEL and CALCTASKDIMS in the *Oracle Essbase Technical Reference*.
- Add the appropriate calculation commands to a calculation script.

  See SET CALCPARALLEL and SET CALCTASKDIMS in the *Oracle Essbase Technical Reference*.

Parallel calculation settings use standard precedence rules:

- The database setting takes precedence over the application setting
- The application setting takes precedence over the server setting.

Setting parallel calculation at the server level enables it for all calculations performed on all applications and databases on the server. You can disable parallel calculation for individual applications or databases by setting parallel calculation at the server level in the configuration file and then adding application-specific or database-specific entries in a calculation script.

Caution!

Read this entire chapter before attempting to enable parallel calculation.

➤ To enable parallel calculation:

1. **If you plan to enable parallel calculation in the configuration file, check the current status to see whether an entry exists.**

   Use the process described in "Checking Current Parallel Calculation Settings" on page 871.

2. **Add or modify CALCPARALLEL in the `essbase.cfg` file on the server, or add SET CALCPARALLEL to a calculation script.**

3. **If needed, enable Essbase to use more than the one sparse dimension to identify tasks for parallel calculation.**

   Use the process described in "Identifying Additional Tasks for Parallel Calculation" on page 872.

4. **If you added entries to the configuration file, restart the server.**

5. **Run the calculation.**

   Oracle recommends that you set the value of CALCPARALLEL to be one less than the number of processors available for calculation. This extra processor can then be used by either the operating system or by the Essbase process responsible for writing out dirty blocks from the cache.

Tip:

You can combine the use of CALCPARALLEL and SET CALCPARALLEL if the site requires it. For example, you can set CALCPARALLEL as off at the server level, and use a calculation script to enable and disable parallel calculation as needed.

Identifying Additional Tasks for Parallel Calculation

By default, Essbase uses the last sparse dimension in an outline to identify tasks that can be performed concurrently. But the distribution of data may cause one or more tasks to be empty; that is, there are no blocks to be calculated in the part of the database identified by a task. This situation can lead to uneven load balancing, reducing parallel calculation effectiveness.

To resolve this situation, you can enable Essbase to use additional sparse dimensions in the identification of tasks for parallel calculation. For example, if you have a FIX statement on a member of the last sparse dimension, you can include the next-to-last sparse dimension from the outline as well. Because each unique member combination of these two dimensions is identified as a potential task, more and smaller tasks are created, increasing the opportunities for parallel processing and improving load balancing.

➤ To increase the number of sparse dimensions used to identify tasks for parallel calculation:

1 **If you are not sure, verify whether parallel calculation is enabled.**

   See "Checking Current Parallel Calculation Settings" on page 871. Without CALCPARALLEL (or SET CALCPARALLEL in a calculation script), CALTASKDIMS has no effect.

2 **Add or modify CALCTASKDIMS in the `essbase.cfg` file on the server, or use the calculation script command SET CALCTASKDIMS at the top of the script.**

   See the *Oracle Essbase Technical Reference*.

3 **If you add or modify CALCTASKDIMS in the `essbase.cfg` file on the server, restart Essbase.**

4 **If you are using a calculation script, run the script.**

Note:

In some cases, Essbase uses fewer dimensions to identify tasks than is specified by CALCTASKDIMS or SET CALCTASKDIMS. See the *Oracle Essbase Technical Reference*.

Monitoring Parallel Calculation

You can view events related to parallel calculation in the application log:

➤ To view the application log, see "Viewing Logs" in the *Oracle Essbase Administration Services Online Help*.

For each calculation pass, Essbase writes several types of information to the application log to support parallel calculation:

- If you have enabled parallel calculation and Essbase has determined that parallel calculation can be performed, Essbase writes a message in the application log:

  ```
  Calculating in parallel with n threads
  ```

  `n` represents the number of concurrent tasks specified in CALCPARALLEL or SETCALCPARALLEL.

- For each formula that prevents parallel calculation (forces serial calculation), Essbase writes a message to the application log:

  ```
  Formula on ((or backward dependence from) mbr memberName prevents
  calculation from running in parallel.
  ```

  `memberName` represents the name of the member where the relevant formula exists. You can look in the application log for such messages and consider removing the formula or, if possible, tagging the relevant member or members as Dynamic Calc so they do not feature in the calculation pass.

- Essbase writes a message to the application log specifying the number of tasks that can be executed concurrently (based on the data, not the value of CALCPARALLEL or SETCALCPARALLEL):

  ```
  Calculation task schedule [576,35,14,3,2,1]
  ```

  The example message indicates that 576 tasks can be executed concurrently. After the 576 tasks complete, 35 more can be performed concurrently, and so on.

  The benefit of parallel calculation is greatest in the first few steps and diminishes as fewer concurrent tasks are performed.

  The degree of parallelism depends on the number of tasks in the task schedule. The greater the number, the more tasks that can run in parallel, and the greater the performance gains.

- Essbase writes a message to the application log indicating how many tasks are empty (contain no calculations):

  ```
  [Tue Jun 27 12:30:44 2007]Local/CCDemo/Finance/essexer/

  Info(1012681) Empty tasks [291,1,0,0,0,0]
  ```

  In the example, Essbase indicates that 291 of the tasks at level 0 were empty.

  If the ratio of empty tasks to the tasks specified in the calculation task schedule is greater than 50% (for example, 291 / 576), parallelism may not be giving you improved performance because of the high sparsity in the data model.

  You can change dense-sparse assignments to reduce the number of empty tasks and increase the performance gains from parallel calculation.

Using Formulas

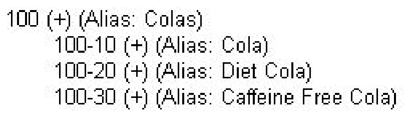
You may achieve significant improvements in calculation performance by carefully using formulas in the database outline. For example, you may achieve improved calculation performance by placing formulas on members in the database outline instead of placing the formulas in a calculation script. See Chapter 22, “Developing Formulas.”

The following sections discuss how to handle formula issues that affect performance.

Consolidating

Using the database outline to roll up values is more efficient than using a formula to calculate values. For example, consider the consolidation on the Sample.Basic database outline in Figure 159.

Figure 159 Consolidation on Sample.Basic Outline



```
100 (+) (Alias: Colas)
    100-10 (+) (Alias: Cola)
    100-20 (+) (Alias: Diet Cola)
    100-30 (+) (Alias: Caffeine Free Cola)
```

Using outline consolidation is more efficient than applying the following formula to the 100 member:

```
100-10 + 100-20 + 100-30
```

Using Simple Formulas

If you use a simple formula, and block size is not unusually large, you can place the formula on a member of either a sparse or a dense dimension without significantly affecting calculation performance. The bigger the block size, the more impact simple formulas have on calculation performance. For a discussion of the relationship between block size and calculation performance, see "Block Size and Block Density" on page 860.

A simple formula is, for example, a ratio or a percentage and meets the following requirements:

- Does not reference values from a different dimension (sparse or dense). For example, a simple formula cannot reference Product -> Jan.
- Does not use range functions. For example, a simple formula cannot use @AVGRANGE, @MAXRANGE, @MINRANGE, or @SUMRANGE.
- Does not use relationship or financial functions. For example, a simple formula cannot use @ANCESTVAL, @NEXT, @PARENTVAL, @SHIFT, @ACCUM, or @GROWTH. For a complete list of relationship and financial functions, see the *Oracle Essbase Technical Reference*.

For information on how formulas affect calculation performance, see "Bottom-Up and Top-Down Calculation" on page 880.

Using Complex Formulas

If you use a complex formula, you can improve performance by applying the following guidelines:

- If possible, apply the formula to a member in a *dense* dimension.
- Use the FIX command in a calculation script to calculate only required data blocks. See "Using the FIX Command" on page 460.
- Increase the density of the database (ratio of existing data blocks to possible data blocks). See "Block Size and Block Density" on page 860.

A complex formula is one that meets any of the following requirements:

- References a member or members in a different dimension (sparse or dense); for example, Product -> Jan.
- Uses one or more range functions, for example, @AVGRANGE, @MAXRANGE, @MINRANGE, or @SUMRANGE.
- Uses relationship or financial functions; for example, @ANCESTVAL, @NEXT, @PARENTVAL, @SHIFT, @ACCUM, or @GROWTH. For a complete list of relationship and financial functions, see the *Oracle Essbase Technical Reference*.

When applied to sparse dimension members, complex formulas create more calculation overhead and therefore slow performance. This problem occurs because the presence of complex formulas requires Essbase to perform calculations on all possible as well as all existing data blocks related to the member with the complex formula. The presence of a relationship or financial function on a sparse dimension member causes Essbase to perform calculations on all blocks, possible as well as existing, increasing the overhead even more.

Thus, a complex formula that includes a relationship or financial function creates a greater overhead increase than does a complex formula that does not include a relationship or financial function.

For a discussion about how complex formulas affect calculation performance, see “Bottom-Up and Top-Down Calculation” on page 880.

Two examples illustrate complex formula overhead:

- If a database has 90 existing data blocks and 100 potential data blocks, the overhead for complex formulas is not large, not more than 10 extra blocks to read and possibly write values to.
- If a database has 10 existing data blocks and 100 potential data blocks, the overhead is as much as ten times what it would be without the complex formula (depending on the outline structure and other factors), as many as 90 extra blocks to read and possibly write to.

In all cases, the lower the ratio of existing data blocks to possible data blocks, the higher the calculation performance overhead and the slower the performance.

Optimizing Formulas on Sparse Dimensions in Large Database Outlines

You can use the SET FRMLBOTTOMUP calculation command to optimize the calculation of formulas in sparse dimensions in large database outlines. With this command, you can force a bottom-up calculation on sparse member formulas that otherwise would be calculated top-down. See “Forcing a Bottom-Up Calculation” on page 881.

Forcing a bottom-up calculation on a top-down formula enables efficient use of the CALC ALL and CALC DIM commands. Review the discussions of the SET FRMLBOTTOMUP calculation command and the CALCOPTFRMLBOTTOMUP configuration setting in the *Oracle Essbase Technical Reference*.

Constant Values Assigned to Members in a Sparse Dimension

If you assign a constant to a member in a sparse dimension, Essbase automatically creates a data block for every combination of sparse dimension members that contains the member.

For example, assume that a member or a calculation script formula contains the following expression:

```
California = 120;
```

In this formula, California is a member in a sparse dimension and 120 is a constant value. Essbase automatically creates all possible data blocks for California and assigns the value 120 to all data cells. Many thousands of data blocks may be created. To improve performance, create a formula that does not create unnecessary values.

➤ To assign constants in a sparse dimension to only those intersections that require a value, use FIX in a manner similar to the following example:

```
FIX(Colas,Misc,Actual)
   California = 120;
ENDFIX
```

In this example, Colas is a member of the sparse dimension, Product; Actual is a member of the dense dimension, Scenario; and Misc is a member of the dense dimension, Measures. The value 120 is assigned to any intersection of California (in the Market dimension), Actual (in the Scenario dimension), Misc (in the Measures dimension), Colas (in the Product dimension), and any member in the Year dimension, because a specific member of Year is not specified in the script.

Because Sample.Basic includes only two sparse dimensions, this example affects only one block. If more sparse dimensions existed, Essbase would ensure data blocks for all combinations of the sparse dimensions with California and Colas, creating blocks if necessary. Within the new blocks, Essbase sets Measures and Scenario values (other than those assigned the value 120) to #MISSING.

Nonconstant Values Assigned to Members in a Sparse Dimension

If you assign nonconstant values to members of a sparse dimension, blocks are created based on the Create Blocks on Equations setting. The Create Blocks on Equations setting is defined at the database level, as a database property. Within calculation scripts, you can temporarily override the Create Blocks on Equations setting. (See "Nonconstant Values" on page 356.)

Consider the effects of the following calculation when West does not have a value and the Create Blocks on Equations setting is ON.

```
West = California + 120;
```

Unneeded blocks may be created for all sparse-member intersections with West, even if the corresponding block value is #MISSING for all of the children of West. Especially in a large database, creation and processing of unneeded blocks requires additional processing time.

To control creation of blocks when you assign nonconstant values to members of a sparse dimension, use the SET CREATEBLOCKONEQ ON | OFF command. The following script includes calculations with this setting off and then on:

```
FIX (Colas);
   SET CREATEBLOCKONEQ OFF
   West = California + 120;
   SET CREATEBLOCKONEQ ON
   East = "New York" + 100;
ENDFIX
```

Because the Create Block on Equation setting is disabled at the beginning, West blocks are created only when values exist for the children of West. Later, because the Create Block on Equation setting is enabled, all blocks for East are created.

Note:

Using SET CREATEBLOCKONEQ affects only creation of blocks during the execution of the calculation script that contains this command. This command does not change the overall database Create Blocks on Equations setting.

For information about using SET CREATEBLOCKEQ ON | OFF in calculation scripts, see the *Oracle Essbase Technical Reference*.

Using Cross-Dimensional Operators

Use caution when using a cross-dimensional operator (->) in the following situations:

On the Left of an Equation

For faster calculation script performance, use FIX in the calculation script to qualify the use of a formula rather than a formula that includes a cross-dimensional operator on the left of an equation.

For example, assume that you want to increase the Jan -> Sales values in Sample.Basic by 5%. To improve performance by calculating only the relevant combinations of members, use the FIX command:

```
FIX(Jan)
   Sales = Sales * .05;
ENDFIX
```

With the FIX command, Essbase calculates the formula only for specified member combinations, in this example, for combinations that include Jan.

Compare this technique to using the slower cross-dimensional operator approach. For the previous example, you place the following formula on the Sales member in the database outline:

```
Sales(Sales -> Jan = Sales -> Jan * .05;)
```

As Essbase cycles through the database, it calculates the formula for every member combination that includes a member from the dimension tagged as time (Jan, Feb, Mar, and so on), although only January combinations need to be calculated.

See "Using the FIX Command" on page 460 and the *Oracle Essbase Technical Reference*.

In Equations in a Dense Dimension

When you use a cross-dimensional operator in an equation in a dense dimension, Essbase does not automatically create the required blocks if both of these conditions apply:

- Resultant values are from a dense dimension.
- The operand or operands are from a sparse dimension.

You can use the following techniques to create the blocks and avoid the performance issue.

- Ensure that the results members are from a sparse dimension, not from a dense dimension. In this example, the results member Budget is from a sparse dimension:

  ```
  FIX(Sales)

      Budget = Actual * 1.1;

  ENDFIX

  FIX(Expenses)

      Budget = Actual *  .95;

  ENDFIX
  ```

- Use the DATACOPY calculation command to create and then calculate the required blocks. See "Using DATACOPY to Copy Existing Blocks" on page 464.
- Use a member formula that contains the dense member equations:

  ```
  FIX(Sales, Expenses)

  Budget (Sales = Sales -> Actual * 1.1;

  Expenses = Expenses -> Actual * .95;)

  ENDFIX
  ```

Managing Formula Execution Levels

Formulas in a block storage outline can have dependencies on one another such that they cause a nested execution of formulas within one or more blocks. Such formulas are called recursive formulas. Sometimes recursive formulas result in large or unending loops that result in abnormal termination of the server.

To avoid abnormal termination, you can use the CALCLIMITEFORMULARECURSION configuration setting to stop a formula execution that reaches 31 execution levels. See the *Oracle Essbase Technical Reference*

Using Bottom-Up Calculation

A top-down calculation is less efficient than a bottom-up calculation, because more blocks are calculated than is necessary. Although a top-down calculation is less efficient than a bottom-up calculation, top-down calculations are necessary in some cases to ensure that calculation results are correct.

The following topics describe which calculation to use in different situations:

Bottom-Up and Top-Down Calculation

Essbase uses one of two calculation methods to do a full calculation of a database outline—bottom-up calculation or top-down calculation. By default, Essbase does a bottom-up calculation.

For a bottom-up calculation, Essbase determines which data blocks must be calculated before it calculates the database. Essbase then calculates only the blocks that must be calculated. The calculation begins with the existing block with the lowest block number and works up through each block in number order until the existing block with the highest block number is reached. See "Block Calculation Order" on page 392.

If the database outline contains a complex member formula, Essbase performs a top-down calculation for the relevant member.

Use the following information to learn more about simple and complex formula interactions with bottom-up and top-down calculation:

Bottom-Up Calculations and Simple Formulas

For simple formulas, Essbase does a bottom-up calculation to determine which blocks must be calculated before running the full calculation. For example, for a simple formula on a member (such as A = B + C), A is calculated only if B or C exists in the database. That is, the dependency of the formula on B and C is known before the calculation is started.

Top-Down Calculations and Complex Formulas

Before starting a calculation, Essbase searches the database outline and marks complex formulas that require top-down calculation; for example, a member formula that contains a cross-dimensional reference. When Essbase reaches a member with a top-down formula, it does a top-down calculation for the member.

When a formula on a member is complex, all possible blocks for the member must be examined to see if an existing block must be changed or a new block created; it is difficult to determine the dependency that blocks have on other blocks before the start of the calculation. The top-down method slows calculation performance because Essbase must search for appropriate blocks to calculate to execute the formula.

When a formula is compiled, if the formula is to be calculated top-down, Essbase logs a message in the application log file.

Consider the following complex formula:

```
A = B -> D + C -> D
```

To calculate the formula, Essbase must examine every combination of A to see whether B -> D or C -> D exists.

See "Using Complex Formulas" on page 875.

Forcing a Bottom-Up Calculation

If it is appropriate for the site, you can force a bottom-up calculation on a top-down formula.

➤ To force a bottom-up calculation, use a tool:

Method	Topic Where Discussed	Location
Calculation function	@CALCMODE in a formula	*Oracle Essbase Technical Reference*
Calculation script command	SET FRMLBOTTOMUP	*Oracle Essbase Technical Reference*
`essbase.cfg` file setting	CALCOPTFRMLBOTTOMUP or CALCMODE	*Oracle Essbase Technical Reference*

Forcing a bottom-up calculation on a formula usually increases performance time. If the formula contains complex functions (for example, range functions) or if the formula's dependencies are not straightforward, a bottom-up calculation may produce results different from those of a top-down calculation.

Caution!

Before changing the setting CALCOPTFRMLBOTTOMUP or using the calculation script command SET FRMLBOTTOMUP in a production environment, check the validity of calculation results by comparing, relative to the same data, the results of a bottom-up calculation and the results of a top-down calculation.

Managing Caches to Improve Performance

The following section describes the caches that are used with block storage databases. For information about the aggregate storage cache, see "Managing the Aggregate Storage Cache" on page 993.

When calculating a database, Essbase uses approximately 30 bytes of memory per member in the database outline. So if the database has 5,000 members, Essbase needs approximately 150 KB of memory to calculate the database.

Note:

You can avoid excess memory use by combining calculation scripts. You can obtain good performance by using parallel calculation with a single calculation script. See "Using Parallel Calculation" on page 866.

Essbase uses memory to optimize calculation performance, especially for large calculations. The amount of memory used is not controllable, except by altering the size of the database outline. However, you can ensure that the memory cache sizes enable Essbase to optimize the calculation.

Essbase uses memory caches to coordinate memory usage:

- Calculator cache. Ensure that the calculator cache is large enough to optimize calculation performance.
- Dynamic calculator cache.
- Index cache. If the database is large, the default index cache is not large enough to provide optimum calculation performance.
- Data cache.
- Data file cache.

Note:

When you first calculate a database, the size of the calculator cache is significant for calculation performance. If possible, ensure that the calculator cache is large enough for Essbase to use the optimal calculator cache option.

See "Sizing Caches" on page 826. Read the entire topic before making changes.

Working with the Block Locking System

When a block is calculated, Essbase locks the block and all blocks that contain the children of the block. Essbase calculates the block and then releases the block and the blocks containing the children.

By default, Essbase locks up to 100 blocks concurrently when calculating a block. This number of block locks is sufficient for most database calculations. If you are calculating a formula in a sparse dimension, Essbase works most efficiently if it can lock all required child blocks concurrently. Therefore, when calculating a formula in a sparse dimension, you may want to set a lock number higher than 100 if you are consolidating very large numbers of children (for example, more than 100). By increasing the number, you ensure that Essbase can lock all required blocks, and performance is not impaired.

Essbase locking behavior depends on the isolation level setting. See "Locking Under Committed Access" on page 782 and "Locking Under Uncommitted Access" on page 784.

Note:

For consolidations in a sparse dimension, block locking is not a consideration, because Essbase does not need to lock all blocks containing children concurrently.

Using SET LOCKBLOCK and CALCLOCKBLOCK

You can use the SET LOCKBLOCK command in a calculation script along with the CALCLOCKBLOCK setting in the `essbase.cfg` file to specify the maximum number of blocks that Essbase can lock concurrently when calculating a block. If you do not modify the default setting, and the default 100 blocks is not sufficient during calculation, the calculation may require more time than expected.

Managing Concurrent Access for Users

Essbase uses the block locking system to manage concurrent access to users. This system ensures that only one user at a time can update or calculate a particular data block. How Essbase handles locking blocks and committing data depends on the isolation level setting.

When Essbase calculates a data block, it creates an exclusive lock; other users cannot update or calculate it, but they can have read-only access. When Essbase finishes the calculation, it releases the block. Other users can then update the block if they have the appropriate security access.

When a user is updating a data block, the block is locked. If a database calculation requires a data block that is being updated by another user, the calculation waits for one of the following conditions:

- For the data block to be released if the isolation level setting is Uncommitted Access.
- For the calculation to complete if the isolation level setting is Committed Access.

Essbase does not provide a message to say that the calculation is waiting for the data block to be released.

You can prevent calculation delays caused by waiting for locked blocks by using Essbase security options to do either of the following:

- Deny access to other users
- Disconnect users from Essbase

For information about security options, see “Disconnecting Users and Terminating Requests in Native Security Mode” on page 639 and “Managing Passwords and User Names in Native Security Mode” on page 640.

For information on how Essbase handles locks and transactions, see “Understanding How Essbase Handles Transactions” on page 786 and “Data Locks” on page 780.

Note:

When Essbase locks a block for calculation, it does not put an exclusive lock on the dependent child blocks, so another user can update values in the child blocks. If necessary, you can use the above security options to prevent such updates.

Using Two-Pass Calculation

You can improve performance significantly by tagging an accounts dimension member as two-pass in the database outline, if it is appropriate for the application. The combination of data and calculation needs may require the use of a calculation script to calculate a formula twice, instead of two-pass tagging, to preserve accuracy.

Use these sections to understand more about two-pass calculation. Decide whether you can tag an accounts dimension member as two-pass to improve performance, or whether you must use a calculation script to calculate a formula twice. This section also provides information about how to enable two-pass calculation or create a calculation script for two-pass calculation.

For information about the interaction of two-pass calculation and attribute members, see Table 18 on page 165.

Understanding Two-Pass Calculation

You can use a two-pass calculation on member formulas that must be calculated twice to produce the correct value.

Whenever possible, Essbase calculates two-pass formulas at the data block level, calculating the two-pass formulas at the same time as the main calculation. Thus, Essbase need not do an extra calculation pass through the database. However, in some situations, Essbase needs an extra calculation pass through the database.

How Essbase calculates the two-pass formulas depends on whether there is a dimension tagged as time as well as a dimension tagged as accounts and on the dense-sparse configuration of the time and account dimensions.

Reviewing a Two-Pass Calculation Example

Consider this calculation required for Profit%:

```
Profit % = Profit % Sales
```

Assume that the following table shows a subset of a data block with Measures and Year as dense dimensions. Measures is tagged as accounts, and Year is tagged as time. The AGGMISSG setting is turned off (the default).

Data values have been loaded into the input cells. Essbase calculates the shaded cells. The numbers in bold show the calculation order for the cells. Cells with multiple consolidation paths are darkly shaded.

Measures -> Year	Jan	Feb	Mar	Qtr1
Profit	75	50	120	**5**
Sales	150	200	240	**6**
Profit%	**1**	**2**	**3**	**4 / 7**

Note:

For information on how cell calculation order depends on database configuration, see "Cell Calculation Order" on page 394.

Essbase uses this calculation order:

1. Essbase calculates the formula `Profit % Sales` for Profit % -> Jan, Profit % -> Feb, Profit % -> Mar, and Profit % -> Qtr1 (1, 2, 3, 4 above).
2. Essbase calculates Profit -> Qtr1 and Sales -> Qtr1 by adding the values for Jan, Feb, and Mar (5, 6 above).
3. Essbase calculates Profit % -> Qtr1 by adding the values for Profit % -> Jan, Profit % -> Feb, and Profit % -> Mar (7 above). This addition of percentages produces the value %125, not the correct result.

Measures/Year	Jan	Feb	Mar	Qtr1
Profit	75	50	120	**245 (5)**
Sales	150	200	240	**590 (6)**
Profit%	**50% (1)**	**25% (2)**	**50% (3)**	**0% (4)** **125% (7)**

4. If you tag Profit % as two-pass in the database outline, Essbase uses the `Profit % Sales` formula to recalculate the Profit % values and produce the correct results.

Measures/Year	Jan	Feb	Mar	Qtr1
Profit	75	50	120	**245 (5)**
Sales	150	200	240	**590 (6)**
Profit%	**50% (1)**	**25% (2)**	**50% (3)**	**0% (4)** **125% (7)** **42% (8)**

For information about multiple calculation passes, see "Calculation Passes" on page 399.

Understanding the Interaction of Two-Pass Calculation and Intelligent Calculation

Two scenarios are described in detail in the following sections. If you are using Intelligent Calculation, use the scenario that matches the configuration of the database; each scenario tells you how to ensure that Essbase accurately calculates two-pass formulas.

These scenarios require that you understand the concepts of Intelligent Calculation. See Chapter 25, "Understanding Intelligent Calculation."

Scenario A

In this scenario, you place formulas in the outline and, as appropriate, tag specific formulas as two-pass for best performance.

No Extra Calculation Pass for Two-Pass Formulas

Because Essbase calculates the two-pass formulas while it is calculating the data block, Essbase need not do an extra calculation pass through the database.

All Data Blocks Marked As Clean

After the calculation, all data blocks are marked as clean for the purposes of Intelligent Calculation.

When you tag a member formula as two-pass in the outline, Essbase does the two-pass calculation while each data block is being calculated. However, when you repeat a formula in a calculation script, Essbase must read the data blocks and write them to memory to recalculate the formula.

Scenario B

In this scenario, you create a calculation script to perform the formula calculation for best performance.

Extra Calculation Pass for Two-Pass Formulas

Essbase calculates the database and then does an extra calculation pass to calculate the two-pass formulas. Even though all data blocks are marked as clean after the first database calculation, Essbase ignores the clean status on the blocks that are relevant to the two-pass formula and recalculates these blocks.

Data Blocks for Two-pass Formulas Not Marked As Clean

After the first calculation, Essbase has marked all data blocks as clean for the purposes of Intelligent Calculation. In a second calculation pass through the database, Essbase recalculates the required data blocks for the two-pass formulas. However, because the second calculation is a partial calculation of the database, Essbase does not mark the recalculated blocks as clean. When you recalculate the database with Intelligent Calculation turned on, these data blocks may be recalculated unnecessarily.

If the database configuration allows Essbase to use Scenario B, consider using a calculation script to perform two-pass formula calculations. If you use a calculation script, Essbase still does an extra calculation pass through the database; however, you can ensure that Essbase has marked all the data blocks as clean after the calculation. See "Creating Calculation Scripts for Two-Pass and Intelligent Calculation" on page 888.

Choosing Two-Pass Calculation Tag or Calculation Script

Although tagging an accounts member as two-pass may bring performance benefits, some applications cannot use this method. Check these qualifications to see whether you should apply a two-pass tag or create a calculation script that performs a calculation twice for best performance and accuracy:

- You can tag a member as two-pass if it is in a dimension tagged as accounts. When you perform a default calculation on the database, Essbase automatically recalculates any formulas tagged as two-pass if they are in the dimension tagged as accounts in the database outline.
- You can tag a member as two-pass if it is a Dynamic Calc or Dynamic Calc and Store member of any dimension. See Chapter 26, "Dynamically Calculating Data Values."
- You may need to use a calculation script to calculate a two-pass formula to obtain accurate results, even if the two-pass tag would provide performance benefits. See "Creating Calculation Scripts for Two-Pass and Intelligent Calculation" on page 888.
- Use a calculation script instead of the two-pass tag to ensure efficient use of Intelligent Calculation. See "Understanding the Interaction of Two-Pass Calculation and Intelligent Calculation" on page 885.
- You must use a calculation script to calculate a formula twice if the database configuration means that Essbase uses Scenario A, as described in "Scenario A" on page 885, and if the formula references values from another data block.
- You may want to use a calculation script to calculate two-pass formulas if the database configuration means that Essbase uses Scenario B, as described in "Scenario B" on page 886.

Enabling Two-Pass on Default Calculations

A database setting enables two-pass calculation in default calculations. When you perform a default calculation on a database with two-pass calculation enabled (the default setting), Essbase automatically attempts to calculate formulas tagged as two-pass in the dimension tagged as accounts in the database outline. This is true even if you have customized the default calculation script.

➤ To perform a default calculation, use a tool:

Tool	Topic	Location
Administration Services	Calculating Block Storage Databases	*Oracle Essbase Administration Services Online Help*
MaxL	**execute calculation**	*Oracle Essbase Technical Reference*
ESSCMD	CALCDEFAULT	*Oracle Essbase Technical Reference*

➤ To enable two-pass calculation, use a tool:

Tool	Topic	Location
Administration Services	Using Two-Pass on a Default Calculation	*Oracle Essbase Administration Services Online Help*
MaxL	**alter database**	*Oracle Essbase Technical Reference*
ESSCMD	SETDBSTATE	*Oracle Essbase Technical Reference*

Creating Calculation Scripts for Two-Pass and Intelligent Calculation

Use these methods to create calculation scripts to perform two-pass calculations with Intelligent Calculation, so that the calculation is accurate and as fast as possible:

- Before the calculation script command that recalculates a two-pass formula, add the SET UPDATECALC OFF command to disable Intelligent Calculation. If you have Intelligent Calculation enabled (the default), Essbase calculates only the data blocks that are not marked as clean, but when you perform a default calculation of the database with Intelligent Calculation enabled, all data blocks are marked as clean, so Essbase does not perform the two-pass formula recalculation.
- When you use a calculation script, Essbase does not automatically recalculate two-pass formulas. Use the CALC TWOPASS command.
- If you have changed the default calculation of CALC ALL, and Intelligent Calculation is enabled, the data blocks may not be marked as clean after the first calculation. See Chapter 25, "Understanding Intelligent Calculation." Also see "Setting Default Calculations" in the *Oracle Essbase Administration Services Online Help*.

To obtain the performance benefits of Intelligent Calculation when performing the first, full calculation of the database, use one of these methods, depending on the calculation needs and outline structure:

- "Intelligent Calculation with a Large Index" on page 889
- "Intelligent Calculation with a Small Index" on page 890
- "Intelligent Calculation Turned Off for a Two-Pass Formula" on page 891

These three options use the following example situation:

The outline has a dimension tagged as accounts, and it is a dense dimension. You want to calculate sales for each product as a percentage of sales for all products. Assume this formula should calculate the dimension:

```
Sales % Sales -> Product
```

When Essbase calculates the data block for each product, it has not yet calculated the value Sales -> Product, so the results for the sales of each product as a percentage of total sales are incorrect.

Intelligent Calculation with a Large Index

If the index is large, and you want to use Intelligent Calculation, you can use any of the following options for the best performance. All three options perform the same tasks.

1. Enable Intelligent Calculation.
2. Calculate the full database and marks the data blocks as clean.
3. Disable Intelligent Calculation.
4. Mark the recalculated blocks as clean, even though this calculation is a partial calculation of the database. If you do not use the command SET CLEARUPDATESTATUS AFTER, Essbase marks data blocks as clean only after a full calculation of the database.
5. Essbase cycles through the database, calculating only the formula for the relevant member (Share of Sales in our example), or calculating all formulas tagged as two-pass in the database outline.

Use a Calculation Script

Use this model to create a calculation script that performs a full calculation of the database with Intelligent Calculation enabled:

```
SET UPDATECALC ON;
CALC ALL;
SET UPDATECALC OFF;
SET CLEARUPDATESTATUS AFTER;
"Share of Sales" = Sales % Sales -> Product;
```

Use a Calculation Script and the Two-Pass Tag

➤ To tag a member as two-pass, and use a calculation script to calculate first the full database, then the two-pass member:

1 **Place a formula in the database outline and tag it as two-pass.**

2 **Place the formula on the appropriate member in the dimension tagged as accounts, in our example, Share of Sales.**

3 **Create a calculation script that performs a full database calculation and then a two-pass calculation:**

```
SET UPDATECALC ON;
CALC ALL;
SET UPDATECALC OFF;
SET CLEARUPDATESTATUS AFTER;
CALC TWOPASS;
```

Use a Client and a Calculation Script

➤ To perform a default calculation from a client and then use a calculation script to perform the formula calculation:

1 **Enable Intelligent Calculation, if this default has been changed.**

2 Perform a full calculation, using any of the tools listed in Table 119 on page 890.

3 Use a calculation script similar to this example to disable Intelligent Calculation and calculate the formula:

```
SET UPDATECALC OFF;
SET CLEARUPDATESTATUS AFTER;
"Share of Sales" = Sales % Sales -> Product;
```

or:

```
SET UPDATECALC OFF;
SET CLEARUPDATESTATUS AFTER;
CALC TWOPASS;
```

Table 119 Methods for Performing a Full Calculation

Tool	Topic	Location
Administration Services	Calculating Databases	*Oracle Essbase Administration Services Online Help*
MaxL	**execute calculation**	*Oracle Essbase Technical Reference*
ESSCMD	CALCDEFAULT	*Oracle Essbase Technical Reference*

See Chapter 25, "Understanding Intelligent Calculation", Chapter 22, "Developing Formulas", and Chapter 28, "Developing Calculation Scripts."

Intelligent Calculation with a Small Index

➤ To use Intelligent Calculation when the index is small:

1 Create a calculation script to calculate the database, but tell Essbase not to mark the calculated data blocks as clean.

2 Mark all data blocks as clean and do not recalculate the data blocks.

```
SET CLEARUPDATESTATUS OFF;
CALC ALL;
CALC TWOPASS;
SET CLEARUPDATESTATUS ONLY;
CALC ALL;
```

With the example script, Essbase performs these tasks:

3 The SET CLEARUPDATESTATUS OFF command tells Essbase not to mark the calculated data blocks as clean.

4 The first CALC ALL command causes Essbase to cycle through the database, calculating all dirty data blocks. Essbase does not mark the calculated data blocks as clean. Essbase does not automatically recalculate the formulas tagged as two-pass in the database outline.

5 The CALC TWOPASS command causes Essbase to cycle through the database, recalculating the formulas that are tagged as two-pass in the dimension tagged as accounts in the database outline. Essbase recalculates the formulas because the required data blocks are not marked as clean by the previous CALC ALL. Essbase does not mark the recalculated data blocks as clean.

6 The SET CLEARUPDATESTATUS ONLY command tells Essbase to mark the data blocks as clean but not to calculate the data blocks. This command disables calculation.

7 **The last CALC ALL command causes Essbase to cycle through the database and mark all the data blocks as clean. Essbase searches the index and marks the data blocks as clean. It does not calculate the data blocks.**

Intelligent Calculation Turned Off for a Two-Pass Formula

➤ To turn Intelligent Calculation off for a Two-Pass formula, create a calculation script that performs these tasks:

- Disables Intelligent Calculation.
- Performs a full calculation.
- Repeats the following two-pass formula:

```
SET UPDATECALC OFF;
CALC ALL;
"Share of Sales" = Sales % Sales -> Product;
```

Choosing Between Member Set Functions and Performance

Queries and calculations that reference a member that has been tagged as Dynamic Calc or Dynamic Calc and Store may be significantly slower than queries and calculations involving the same members, if the member has formulas involving any of these functions:

- @CURRMBR
- @PARENT
- @SPARENTVAL
- @ANCEST
- @SANCESTVAL

If you are experiencing slow performance, consider either removing the dynamic calculation tag or removing these functions from the attached formula.

Using Reference Cubes to Improve @XREF Performance

The @XREF function pulls information from one database to another; for example, to get inflation rates from a different database than the database containing the sales data to be calculated. (See the *Oracle Essbase Technical Reference*.) Accessing data across databases in different applications can involve multiple data transfers, increasing performance time. Depending on size characteristics of the referenced database, using reference cubes can decrease data transfer time.

Understanding Reference Cubes

The @XREF function is executed in a calculation on the target database. The @XREF syntax identifies the source database containing the desired information, and a member list statement that qualifies what member information is needed from the source database. When reference cubes are created, a copy of the source database is copied to the memory with the target (requesting) database. By associating the information with the target database, the number of data transfers is minimized, improving performance.

Depending on the volatility of the source data values and how @XREF is used, two types of reference cubes are available: active and passive. A creation parameter defines a reference cube as active or passive.

Use active reference cubes when the source data is relatively stable or when @XREF is included in dynamic calculations on the target database. When an active reference cube is loaded to the target (written to memory), the reference cube is registered with the source database so that source database changes can update reference cube values dynamically. Unloading or deleting a reference cube deregisters it.

Passive reference cubes are not registered, because the source database does not dynamically update passive reference cubes with value changes. Use passive reference cubes when source data values change frequently or when the @XREF function is used in batch calculations on the target database. Once a passive reference cube has been created and loaded, it gets refreshed at the start of every batch calculation.

Note:

If an @XREF function associated with a dynamically calculated target member references a passive reference cube, the reference cube is ignored, and the @XREF request goes directly to the source database.

The following features and situations do not support reference cubes:

- Aliases. Use actual member names, not aliases.
- Duplicate member names
- Dynamic Time Series members
- The presence of attribute dimensions on the source database
- LROs within the reference cube data
- Cascading changes through a series of active reference cubes. For example, database A uses database B as a reference cube source. Database B uses database C as a reference cube source. Change updates on data from database C update database B reference cubes. However, if the updated values on database B affect a reference cube on database A, database A is not updated.

A reference cube probably will not improve @XREF performance when the referenced database has any of the following characteristics:

- Any dimension contains more than 100 members
- More than four or five dimensions exist
- Total database contains more than 1,000,000 cells

- The referenced slice is less than 10%–20% of the referenced cube size (applies to passive reference cubes)

Working with Reference Cubes

Before a reference cube can be used, it must be created and loaded. Creating a reference cube defines the source database and member content of the reference cube. Creating a reference cube also loads it. You can unload a reference cube and reload it. Loading a reference cube retrieves reference cube values, placing them in target memory.

You can display information about active reference cubes on the target or about reference cubes registered on a particular source.

➤ To manage reference cubes, use MaxL statements (details in the *Oracle Essbase Technical Reference*).

Action	MaxL Statement
Create a reference cube	**create passive|active reference_cube**
Delete a reference cube	**drop reference_cube**
Load or unload a reference cube	**alter database**
List active reference cube information	**display reference_cube**
List information about registered reference cubes	**display reference_cube_reg**

Note:

If a database works with many reference cube registrations, calculation time may be improved by using MaxL `alter database` to increase the implicit commit row or block settings.

Sizing Reference-Cube Memory Impact

Individual reference cubes are the size of the source database. Because databases can grow, when you create a reference cube you must specify a maximum size. Databases can have multiple reference cubes. The REFERENCECUBESIZELIMIT configuration setting uses the following syntax to specify the maximum memory to be set aside for all the reference cubes loaded at the same time for a specific database:

REFERENCECUBESIZELIMIT *TargetApp TargetDB "max_cell_k_count_size"*

For example:

REFERENCECUBESIZELIMIT Sample.Basic 100

All sizes are specified in terms of 1000 cells. To calculate the number of cells for a database, you must know the number of cells in an expanded block (including stored and dynamic members) and the number of potential blocks. The number of cells in an expanded block is the product of the number of members across the dense dimensions. Multiply this product by the potential number of blocks (shown on the Database Statistics tab of the Database Properties window in Administration Services Console). For a database with multiple reference cubes, sum the cell counts of its reference cubes. Divide the cell count by 1000 for the number to be used in the REFERENCECUBESIZELIMIT configuration setting and the create passive|active reference cube MaxL statement. The default size for REFERENCECUBESIZELIMIT is 8 (8000 cells).

During operations, when the sum of the cell-count sizes of multiple reference cubes reaches the maximum set by the configuration setting, no more reference cubes are loaded.

Consolidating #MISSING Values

If no data value exists for a combination of dimension members, Essbase gives the combination a value of #MISSING. Essbase treats #MISSING values and zero (0) values differently.

Understanding #MISSING calculation

Table 120 shows how Essbase calculates #MISSING values. In this table, X represents any number:

Table 120 How Essbase Treats #MISSING Values

Calculation/Operation	Result
`X + #MISSING`	`X`

`X – #MISSING`
`#MISSING – X`
	`X`
	`-X`

`X * #MISSING`	`#MISSING`

`X / #MISSING`
`#MISSING / X`
`X / 0`
	`#MISSING`
	`#MISSING`
	`#MISSING`

`X % #MISSING`
`#MISSING % X`
`X % 0`
	`#MISSING`
	`#MISSING`
	`#MISSING`

`X == #MISSING`	`FALSE, unless X is #MISSING`

`X != #MISSING`
`X < > #MISSING`
	`TRUE, unless X is #MISSING`
	`TRUE, unless X is #MISSING`

`X <= #MISSING`	`(X <= 0)`
`X >= #MISSING`	`(X >= 0) or (X == #MISSING)`
`X > #MISSING`	`(X > 0)`
`X < #MISSING`	`(X < 0)`

Calculation/Operation	Result
`X AND #MISSING:`	
`Y AND #MISSING, where Y represents any nonzero value`	`#MISSING`
`0 AND #MISSING`	`0`
`#MISSING AND #MISSING`	`#MISSING`
`X OR #MISSING:`	
`Y OR #MISSING, where Y represents any nonzero value`	`Y`
`0 OR #MISSING`	`#MISSING`
`#MISSING OR #MISSING`	`#MISSING`
`IF (#MISSING)`	`IF (0)`
f `(#MISSING)`	`#MISSING for any Essbase function of one variable`
f `(X)`	`#MISSING for any X not in the domain of` *f* `and any Essbase function of more than one variable (except where specifically noted)`

By default, Essbase does not roll up #MISSING values. However, if you always load data at level 0 and never at parent levels, you should enable the setting for consolidating #MISSING values. This setting provides a calculation performance improvement of 1%–30%. The performance improvement varies, depending on database size and configuration.

Caution!

The default, not consolidating #MISSING values, must be in effect if you load data at parent, rather than child, levels, if any child member combinations have #MISSING values. If all child member combinations have any other values, including zero (0), Essbase rolls up the child values and overwrites the parent values correctly, so you can safely change the default.

Changing Consolidation for Performance

To consolidate, enable the setting for consolidating #MISSING values by using one of the methods described above. The degree of performance improvement you achieve depends on the ratio between upper-level blocks and input blocks in the database.

➤ To change how Essbase consolidates #MISSING values, use a tool:

Tool	Topic	Location
Administration Services	Aggregating Missing Values During Calculation	*Oracle Essbase Administration Services Online Help*
Calculation script	SET AGGMISSG	*Oracle Essbase Technical Reference*
MaxL	**alter database**	*Oracle Essbase Technical Reference*
ESSCMD	SETDBSTATEITEM	*Oracle Essbase Technical Reference*

Note:

If you enable the setting for consolidating #MISSING values, the cell calculation order within a data block changes. See "Cell Calculation Order" on page 394.

When the setting for consolidating #MISSING values is disabled, note that the performance overhead is particularly high in the following situations:

- When the ratio of calculated data blocks to input data blocks is low
- When you load many data values at parent levels on sparse dimensions; for example, in the Sample.Basic database, if you load many data values into East in a sparse Market dimension

In these situations, the performance overhead is 10%–30%. If calculation performance is critical, you may want to reconsider the database configuration or how you load data.

For a information on how Essbase calculates #MISSING values, see "Consolidating #MISSING Values" on page 894.

Removing #MISSING Blocks

You can use the CLEARDATA command to change the value of cells in a block to #MISSING. It does not remove the data blocks. These extra blocks can slow retrieval and calculation performance.

If the #MISSING blocks are slowing performance, perform either action:

- Use the CLEARBLOCK command to remove the data blocks.
- Export the data and re-import it (see the *Oracle Hyperion Enterprise Performance Management System Backup and Recovery Guide*).

Note:

Removing empty blocks improves performance when data values already have been loaded. However, data load process time increases if new values require that blocks be created.

Identifying Additional Calculation Optimization Issues

The relationship between calculation and performance is also described in the following chapters:

- In Chapter 26, “Dynamically Calculating Data Values,” see the following topics:
  - “Benefitting from Dynamic Calculation” on page 419
  - “Choosing Between Dynamic Calc and Dynamic Calc and Store” on page 423
  - “Reducing the Impact on Retrieval Time” on page 428
  - “Dynamically Calculating Data in Partitions” on page 433
- In Chapter 28, “Developing Calculation Scripts,” see the following topics:
  - “Specifying Global Settings for a Database Calculation” on page 450
  - “Writing Calculation Scripts for Partitions” on page 466

For the relationship of two-pass calculation and the SET CLEARUPDATESTATUS command, see the *Oracle Essbase Technical Reference*.

When you convert currencies using the CCONV command, the resulting data blocks are marked as *dirty* for the purposes of Intelligent Calculation. This means that Essbase recalculates all the converted blocks when you recalculate the database. See Chapter 25, “Understanding Intelligent Calculation.”

56 Optimizing Reports and Other Types of Retrieval

In This Chapter

Changing Buffer Size

The time required to generate a report varies depending upon factors such as the size of the database you are reporting from, the number of queries included in the script, and the size of the report buffer.

Configurable variables specify the size of the buffers used for storing and sorting data extracted by retrievals. The buffer should be large enough to prevent unnecessary read and write activities.

See "Report Extractor" on page 508.

The following report variables are used in the conditional retrieval and data sorting commands.

Setting the Retrieval Buffer Size

The database retrieval buffer is a server buffer, per database, that holds extracted row data cells before they are evaluated. Retrieval tools such as the Essbase Retrieval Wizard and the Report Writer use this buffer.

When the retrieval buffer is full, the rows are processed and the buffer is reused. If this buffer is too small, frequent reuse of the area can increase retrieval times. If this buffer is too large, too much memory may be used when concurrent users perform queries, also increasing retrieval times.

The following sections describe ways you can manage retrieval buffer sizes.

Manually Setting the Retrieval Buffer Size

Each database has a retrieval buffer setting, in kilobytes, that you can change. The default buffer size is 10 KB for 32-bit platforms and 20 KB for 64-bit platforms. If you are increasing the size of the buffer, Oracle recommends that you do not exceed 100 KB, although the size limit is set at 100,000 KB.

➤ To manually set the retrieval buffer size, use a tool:

Tool	Topic	Location
Administration Services	Setting the Size of Retrieval Buffers	*Oracle Essbase Administration Services Online Help*
MaxL	**alter database**	*Oracle Essbase Technical Reference*
ESSCMD	SETDBSTATEITEM	*Oracle Essbase Technical Reference*

Note:

If an outline does not include Dynamic Calc, Dynamic Times Series, or attribute members, using the VLBREPORT configuration parameter to dynamically determine the size of the retrieval buffer overrides the database retrieval buffer size setting. See "Enabling Dynamic Retrieval Buffer Sizing" on page 900.

Enabling Dynamic Retrieval Buffer Sizing

If a database has large block size and retrievals include a large percentage of cells from each block across several blocks, consider setting the VLBREPORT option to TRUE in the Essbase configuration file `essbase.cfg`.

When the VLBREPORT setting is TRUE, Essbase internally determines an optimized retrieval buffer size for reports that access more than 20% of the cells in each block across several blocks. This setting takes effect only if the outline does not include Dynamic Calc, Dynamic Times Series, or attribute members. The VLBREPORT configuration setting overrides the manually set retrieval buffer size.

Setting VLBREPORT to TRUE can result in faster query response times for concurrent and serial queries at the cost of a slight increase in memory required for each query.

Setting the Retrieval Sort Buffer Size

The retrieval sort buffer size setting specifies the size, in kilobytes, of the server buffer that holds the data to be sorted during a Oracle's Hyperion® Essbase® Spreadsheet Toolkit or Report Writer retrieval. If the sort buffer is full, Essbase posts an error message.

Essbase requires a larger retrieval sort buffer size on 64-bit platforms than on 32-bit platforms. If you encounter an error indicating that the retrieval sort buffer limit has been exceeded, increase the setting by a factor of two.

You can adjust the buffer size on a per-database basis. The default buffer size is set to 10 KB on 32-bit platforms and to 20 KB on 64-bit platforms.

➤ To set the retrieval sort buffer size, use a tool:

Tool	Topic	Location
Administration Services	Setting the Size of Retrieval Buffers	*Oracle Essbase Administration Services Online Help*
MaxL	**alter database**	*Oracle Essbase Technical Reference*
ESSCMD	SETDBSTATEITEM	*Oracle Essbase Technical Reference*

Setting Numeric Precision

The NUMERICPRECISION setting, used by the RESTRICT command, defines the number of precision digits the internal numerical comparison considers in the Report Extractor. If a precision setting is greater than necessary for the data, retrieval is slower than it could be. Identify the correct precision level and adjust NUMERICPRECISION accordingly.

Table 121 lists the setting that you specify in `essbase.cfg` on the server to set the NUMERICPRECISION. If you change an `essbase.cfg` setting, restart Essbase Server to apply the change.

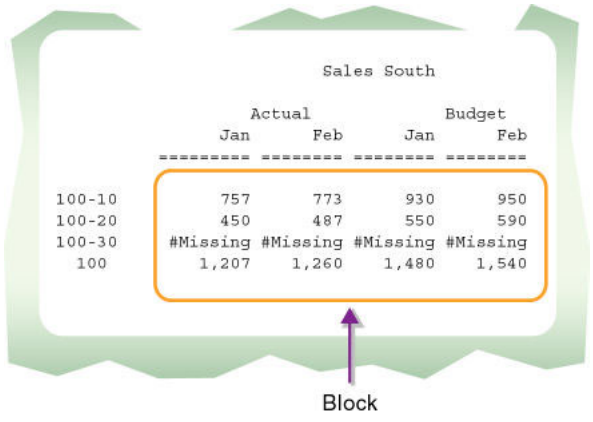
Table 121 Setting Messages in the Server Using essbase.cfg

Setting	Definition	For More Information
NUMERICPRECISION	An `essbase.cfg` setting that determines the number of precision digits used by Report Extractor	*Oracle Essbase Technical Reference*

Generating Symmetric Reports

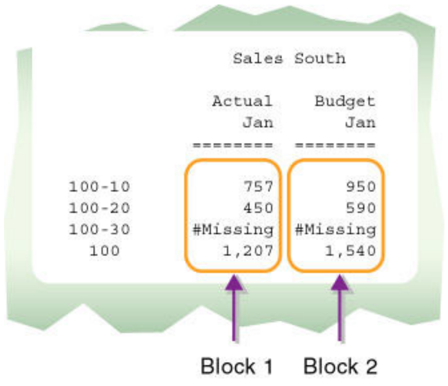
If report processing time is of primary importance, and you are using Report Writer, consider making all reports symmetric. Symmetric reports provide better processing performance than asymmetric reports, because the Report Extractor, in one pass, composes the member list based on all possible member combinations. With asymmetric reports, the Extractor must retrieve and process each block of possible member combinations separately.

Figure 160   Symmetric Report Member Combinations Supporting One Pass



```
                    Sales South

             Actual                  Budget
          Jan      Feb         Jan       Feb
========= ======== ======== ========
100-10      757      773      930      950
100-20      450      487      550      590
100-30 #Missing #Missing #Missing #Missing
100       1,207    1,260    1,480    1,540
```

Figure 161   Asymmetric Report Member Combinations Requiring Multiple Passes



```
               Sales South

           Actual   Budget
              Jan      Jan
         ======== ========
100-10        757      950
100-20        450      590
100-30   #Missing #Missing
100         1,207    1,540
```

See “Report Extractor” on page 508.

Improving Retrieval Performance on Large Dimensions

Queries on large dimensions often have large resource requirements. However, these queries typically are sparse, meaning that the number of nonempty values returned is small compared to the size of the input query. For these large but sparse queries, we suggest using the following special MDX and Report Writer functions to help Essbase more efficiently use memory and processor resources. These functions optimize retrieval performance by attempting to handle only nonempty combinations.

- Leaves() MDX function
- NonEmptySubset() MDX function
- MDX optimization properties: NONEMPTYMEMBER and NONEMPTYTUPLE
- Leaves Report Writer Command
- Generation or level specification in Descendants and Idescendants Report Writer commands (when used within Link command)

Organizing Members to Optimize Data Extraction

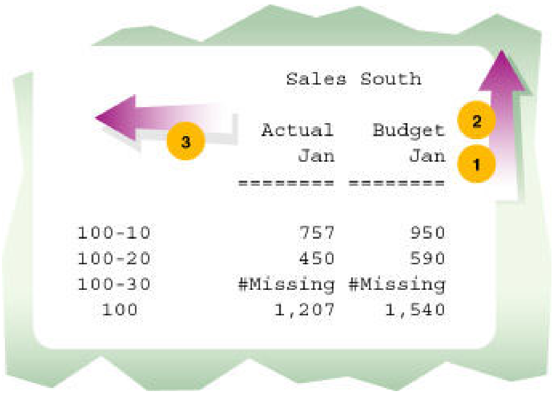
Report Extractor extracts data in a certain order for Report Writer. If you do not require a formatted report and you are using Report Writer, you can reduce the time required to generate the report by using either of these strategies:

- Creating the report script in the same order as Report Extractor extracts data
- Grouping dense dimensions in columns and grouping sparse dimensions in rows

These strategies save the most time if used to create large production reports.

Report Extractor looks at data from bottom to top and right to left, starting from the bottom column member to the top column member and proceeding from the innermost row member (right) to the outermost row member (left). Figure 162 illustrates the sequence in which the report is read.

Figure 162 How Report Extractor Examines Data



```
                     Sales South

                  Actual   Budget
                     Jan      Jan
                ======== ========

100-10               757      950
100-20               450      590
100-30          #Missing #Missing
  100              1,207    1,540
```

The column members come from dense dimensions and the row members from sparse dimensions. To reduce the time to extract data, group dense dimensions first, then group sparse dimensions in the sequence in which they are displayed in the outline.

When dense dimensions are nested in the report columns, Report Extractor examines each data block only once, improving performance time.

Because attributes are sparse dimensions and are dynamically calculated, Essbase cannot use the sparse data extraction method when a report contains attribute dimensions.

Understanding Reports for Outlines That Contain Dynamic or Transparent Members

If you generate a report that accesses a database outline that contains Dynamic Calc and Store members, the first time that you generate the report takes longer than subsequent retrievals that access the same data block.

If you generate a report that accesses a database outline that contains Dynamic Calc or Dynamic Time Series members, Essbase calculates the member every time a report is generated, increasing the reporting time.

See Chapter 26, “Dynamically Calculating Data Values.”

If you run a report that contains transparent members, the report takes longer to generate, because it must access more than one server to retrieve the required data.

Limiting LRO File Sizes

You can limit the size of files that users can link to a database. Limiting the size prevents users from taking up too much of the server resources by storing extremely large objects. See “Limiting LRO File Sizes for Storage Conservation” on page 186.

Part X

Creating, Calculating, and Managing Aggregate Storage Databases

In Creating, Calculating, and Managing Aggregate Storage Databases:

57 Comparison of Aggregate and Block Storage

In This Chapter

Introduction

Essbase provides an aggregate storage kernel as a persistence mechanism for multidimensional databases. Aggregate storage databases enable dramatic improvements in both database aggregation time and dimensional scalability. The aggregate storage kernel is an alternative to the block storage kernel. Aggregate storage databases typically address read-only, “rack and stack” applications that have large dimensionality, such as the following applications:

- Customer analysis. Data is analyzed from any dimension, and there are potentially millions of customers.
- Procurement analysis. Many products are tracked across many vendors.
- Logistics analysis. Near real-time updates of product shipments are provided.

A sample application (ASOsamp), a data file, and a rules file are provided to demonstrate aggregate storage functionality.

Aggregate storage applications, which differ from block storage applications in concept and design, have limitations that do not apply to block storage applications. The following sections describe the differences.

Inherent Differences

Table 122 Inherent Differences Between Aggregate Storage and Block Storage

Inherent Differences	**Aggregate Storage**	**Block Storage**
Storage kernel	Architecture that supports rapid aggregation, optimized to support high dimensionality and sparse data	Multiple blocks defined by dense and sparse dimensions and their members, optimized for financial applications
Physical storage definition	Through the Application Properties window, Tablespaces tab in Administration Services	Through the Database Properties window, Storage tab in Administration Services
Create database	Migrate a block storage outline or define after application creation **Note:** Do not use the file system to copy a block storage outline into an aggregate storage application. Use the migration wizard in Administration Services to migrate the outline.	Define after application creation
Copy database	Not supported	Supported
Databases supported per application	One	Several (one recommended)
Application and database names	See "Naming Restrictions for Applications and Databases" on page 1059. Names reserved for tablespaces, cannot be used as application or database names: ● default ● log ● metadata ● temp	See "Naming Restrictions for Applications and Databases" on page 1059.
Application and database information display	Displayed in the Application Properties window and the Database Properties window in Administration Services. (Information not supported by or relevant to aggregate storage applications is not shown. For a description of aggregate storage-specific information, see the *Oracle Essbase Administration Services Online Help* for the Application Properties window and Database Properties window.)	Displayed in the Application Properties window and the Database Properties window in Administration Services
Configuration settings (`essbase.cfg`)	For a list of the settings that apply to aggregate storage databases, see the *Oracle Essbase Technical Reference*.	For a list of the settings that do not apply to block storage databases, see the *Oracle Essbase Technical Reference*.

Outline Differences

Table 123 Outline Differences Between Aggregate Storage and Block Storage

Outline Functionality	**Aggregate Storage**	**Block Storage**
Dense or sparse dimension designation	Not relevant	Relevant
Multiple hierarchies enabled, dynamic hierarchy, or stored hierarchy designation	Relevant	Irrelevant
Accounts dimensions and members on dynamic hierarchies	Support with the following exceptions: ● No two-pass calculation (however, for information on specifying the calculation order, see "Calculation Order" on page 973) ● No association of attribute dimensions with the dimension tagged Accounts ● Additional restrictions for shared members. See "Alternate Hierarchies" on page 920.	Full support
Members on stored hierarchies	Support with the following exceptions: ● Support for the ~ (no consolidation) operator (underneath label-only members only) and the + (addition) operator ● Cannot have formulas ● Restrictions on label only members (See Member storage types.) ● No Dynamic Time Series members ● Stored hierarchy dimensions cannot have shared members. Stored hierarchies within a multiple hierarchies dimension can have shared members. See "Stored Hierarchies" on page 918.	Full support
Member storage types	Support with the following exceptions: ● Dynamic Calc and Store not relevant ● On stored hierarchies, two limitations if a member is label only: ❍ All dimension members at the same level as the member must be label only ❍ The parents of the member must be label only. **Note:** On dynamic hierarchies, ability to tag any member as label only **Note:** On conversion from a block storage database, attribute dimension members are tagged as Dynamic Calc. On standard dimension members Dynamic Calc tags are converted and tagged as stored members,	Support for all member storage types in all types of dimensions except attribute dimensions

Outline Functionality	Aggregate Storage	Block Storage
	which changes the Members Stored value on the Dimensions tab of the Database Properties window in Administration Services.	
Ragged hierarchies and hierarchies with more than 10 levels	Support, with possible performance impact	Support
Outline validation	● When database is started ● When outline is saved ● When block storage outline is converted to aggregate storage outline ● When user requests	● When outline is saved ● When user requests
Outline paging	Support	No support
Database restructure	Levels of restructure; see "Aggregate Storage Database Restructuring" on page 994.	Levels of restructure; see Chapter 53, "Optimizing Database Restructuring".

Calculation Differences

Table 124 Calculation Differences Between Aggregate Storage and Block Storage

Calculation Functionality	Aggregate Storage	Block Storage
Database calculation	Aggregation of the database, which can be predefined by defining aggregate views	Calculation script or outline consolidation
Formulas	Allowed with the following restrictions: ● Must be valid numeric value expressions written in MDX (cannot contain % operator, replace with expression: (*value1* / *value2*) * 100) ● No support for Essbase calculation functions ● On dynamic hierarchy members, formulas are allowed without further restrictions	Support for Essbase calculation functions
Calculation scripts	Not supported	Supported
Attribute calculations dimension	Support for Sum	Support for Sum, Count, Min, Max, and Average
Calculation order	Member formula calculation order can be defined by the user using the solve order member property	Defined by the user in the outline consolidation order or in a calculation script

Partitioning Differences

Table 125 Partitioning Differences Between Aggregate Storage and Block Storage

Partitioning Functionality	Aggregate Storage	Block Storage
Partitioning	Support with the following restrictions: ● No outline synchronization	Fully supported

Data Load Differences

Table 126 Data Load Differences Between Aggregate Storage and Block Storage

Data Load Functionality	Aggregate Storage	Block Storage
Cells loaded through data loads	Only level 0 cells whose values do not depend on formulas in the outline are loaded	Cells at all levels can be loaded (except Dynamic Calc members)
Update of database values	At the end of a data load, if an aggregation exists, the values in the aggregation are recalculated	No automatic update of values. To update data values, you must execute all necessary calculation scripts.
Data load buffers	The loading of multiple data sources into aggregate storage databases is managed through temporary data load buffers.	Not supported
Atomic replacement of the contents of a database	When loading data into an aggregate storage database, you can replace the contents of the database or the contents of all incremental data slices in the database.	Not supported
Data slices	Aggregate storage databases can contain multiple slices of data. Data slices can be merged.	Not supported
Dimension build for shared members	Full support for parent-child build method. Duplicate generation (DUPGEN) build method limited to building alternate hierarchies up to generation 2 (DUPGEN2).	Support for all build methods
Loading data mapped to dates	In a date-time dimension, you can load data into level-0 members using supported date-format strings instead of member names.	Date-time dimension type is not supported.

Query Differences

Table 127 Query Differences Between Aggregate Storage and Block Storage

Query Functionality	Aggregate Storage	Block Storage
Report Writer	Supported, except for commands related to sparsity and density of data	Fully supported
Spreadsheet Add-in	Supported, with limited ability to change data (write-back)	Fully supported
API	Supported	Supported
Export	Support with the following restrictions: ● Export of level 0 data only (no upper-level export) ● No columnar export	Supported
MDX queries	Supported	Supported
Queries on attribute members that are associated with non-level 0 members	Returns values for descendants of the non-level 0 member. (See also "Design Considerations for Attribute Queries" on page 923.)	Returns #MISSING for descendants of the non-level 0 member
Queries on attribute members and shared members	A shared member automatically shares the attribute associations of its nonshared member	A shared member does not share the attribute associations of its nonshared member
Query logging	Not supported	Supported
Query performance	Considerations when querying data from a dimension that has multiple hierarchies. See "Query Design Considerations for Aggregate Storage" on page 924.	Hierarchies not relevant

Feature Differences

Table 128 Feature Differences Between Aggregate and Block Storage

Features	Aggregate Storage	Block Storage
Aliases	Supported	Supported
Currency conversion	Not supported	Supported
Data mining	Not supported	Supported
Hybrid analysis	Support with the following restriction: queries that contain a relational member and an Essbase member with a formula in the same query are not supported.	Supported

Features	Aggregate Storage	Block Storage

	For example, if California is a relational member, and the member Profit has a formula, the following report script returns an error:
	`Jan`
	`California`
	`Profit`
	`!`	

Incremental data load	Supported	Supported
LROs	Not supported	Supported
Time balance reporting	Support with the following restrictions: ● Skip Zeros is not supported ● Time dimension must contain at least one stored hierarchy ● Shared members must be at level zero	Supported
Triggers	After-update triggers supported	On-update triggers and after-update triggers supported
Unicode	Supported	Supported
Variance reporting	Not supported	Supported
Date-time dimension type and linked attribute dimensions	Supported	Not supported
User ability to change data (write-back)	Transparent partition technique used to enable limited write-back	Fully supported

58 Aggregate Storage Applications, Databases, and Outlines

In This Chapter

The information in this chapter applies to aggregate storage applications, databases, and outlines and includes information on how these processes differ from block storage processes.

Also see:

- Chapter 57, "Comparison of Aggregate and Block Storage"
- "Creating Applications and Databases" on page 115
- "Creating and Changing Database Outlines" on page 127
- "Setting Dimension and Member Properties" on page 141
- "Working with Attributes" on page 159

Introduction

Aggregate storage applications and databases and block storage applications and databases differ in concept and design. Some block storage outline features do not apply to aggregate storage. For example, the concept of dense and sparse dimensions does not apply. See Chapter 57, "Comparison of Aggregate and Block Storage".

A new sample application (ASOsamp), a data file, and a rules file are provided to demonstrate aggregate storage functionality.

Process for Creating Aggregate Storage Applications

This topic provides a high-level process for creating an aggregate storage application.

1. Create an aggregate storage application, database, and outline.

   See "Creating Aggregate Storage Applications, Databases, and Outlines" on page 916.

2. Use tablespaces to optimize data storage and retrieval.

   See "Managing Storage for Aggregate Storage Applications" on page 991.

3. Specify the maximum size of the aggregate storage cache.

   See "Managing the Aggregate Storage Cache" on page 993.

4. Load data into the aggregate storage database. A data load can be combined with a dimension build.

   See "Preparing Aggregate Storage Databases" on page 954. You can preview a subset of the data in Administration Services. See "Previewing Data" in the *Oracle Essbase Administration Services Online Help*.

5. Precalculate chosen aggregations to optimize retrieval time.

   See "Calculating Aggregate Storage Databases" on page 971.

6. View database statistics.

   See "Viewing Aggregate Storage Statistics" in the *Oracle Essbase Administration Services Online Help*.

7. If required, enable write-back by using the Aggregate Storage Partition Wizard.

   See "Using a Transparent Partition to Enable Write-Back for Aggregate Storage Databases" on page 936.

8. View data using Oracle Hyperion tools (for example Spreadsheet Add-in) or third-party tools.

Creating Aggregate Storage Applications, Databases, and Outlines

You must create an aggregate storage application to contain an aggregate storage database. An aggregate storage application can contain only one database. You can create an aggregate storage application, database, and outline in the following ways:

- Convert a block storage outline to an aggregate storage outline, and create an aggregate storage application to contain the converted database and outline.
- Create an aggregate storage application and database. The aggregate storage outline is created automatically when you create the database.

Note:

An aggregate storage outline cannot be converted to a block storage outline.

For information on loading dimensions and members into an aggregate storage outline, see "Building Dimensions in Aggregate Storage Databases" on page 955 and "Loading Data into Aggregate Storage Databases" on page 957.

Aggregate storage application and database information differs from block storage information, and specific naming restrictions apply to aggregate storage applications and databases. See Table 122.

➤ To convert a block storage outline to an aggregate storage outline, use a tool:

Tool	Topic	Location
Administration Services	Aggregate Storage Outline Conversion Wizard	*Oracle Essbase Administration Services Online Help*
MaxL	**create outline**	*Oracle Essbase Technical Reference*

Note:

Do not use the file system to copy a block storage outline into an aggregate storage application. Use the Aggregate Storage Outline Conversion Wizard in Administration Services to convert the outline.

➤ To create an aggregate storage application or database, use a tool:

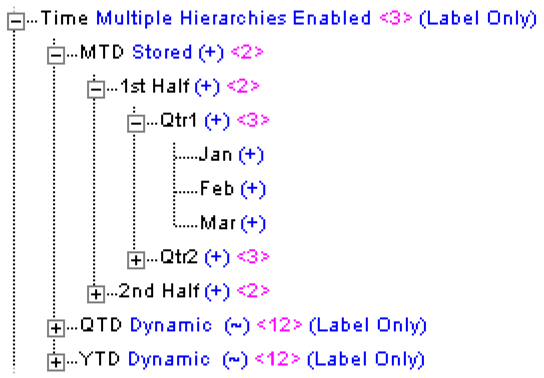
Tool	Topic	Location
Administration Services	Creating Applications Creating Databases	*Oracle Essbase Administration Services Online Help*
MaxL	**create application** **create database**	*Oracle Essbase Technical Reference*

When creating aggregate storage applications, databases, and outlines, consider the differences between aggregate storage and block storage and issues specific to aggregate storage. See the following sections, and also see Chapter 59, "Aggregate Storage Time-Based Analysis."

Hierarchies

In aggregate storage outlines and block storage outlines, dimensions are structured to contain one or more hierarchies of related levels and members within the levels. For example, the Time dimension in the ASOsamp.Sample database (see Figure 163) includes the hierarchies MTD, QTD, and YTD.

Figure 163 ASOSamp.Sample Database Outline Showing Multiple Hierarchies and Members on the Time Dimension



```
Time Multiple Hierarchies Enabled <3> (Label Only)
    MTD Stored (+) <2>
        1st Half (+) <2>
            Qtr1 (+) <3>
                Jan (+)
                Feb (+)
                Mar (+)
            Qtr2 (+) <3>
        2nd Half (+) <2>
    QTD Dynamic (~) <12> (Label Only)
    YTD Dynamic (~) <12> (Label Only)
```

In an aggregate storage database, you can create two types of hierarchies:

- Stored
- Dynamic

The two types of hierarchies have different advantages and restrictions. A dimension may contain both types of hierarchies. To use multiple hierarchies in a dimension (even if they are all stored hierarchies), you must enable multiple hierarchies for that dimension.

➤ To enable multiple hierarchies for a dimension, tag the dimension member as multiple hierarchies enabled using a tool:

Tool	**Topic**	**Location**
Administration Services	Defining Hierarchies in Aggregate Storage Outlines	*Oracle Essbase Administration Services Online Help*
MaxL	**import database**	*Oracle Essbase Technical Reference*

When you tag a dimension member as multiple hierarchies enabled, it is automatically tagged label only.

If you do not tag the dimension as multiple hierarchies enabled, Essbase automatically tags the dimension as a stored hierarchy (except the dimension tagged as Accounts, which is automatically tagged as a dynamic hierarchy).

Note:

The first hierarchy in a multiple hierarchies enabled dimension must be a stored hierarchy.

Stored Hierarchies

Members of stored hierarchies are aggregated according to the outline structure. Because aggregate storage databases are optimized for aggregation, the aggregation of data values for stored hierarchies is very fast. To allow this fast aggregation, members of stored hierarchies have the following restrictions:

- Stored hierarchies can have the no-consolidation (~) operator (only underneath label only members) or the addition (+) operator.
- Stored hierarchies cannot have formulas.

Stored hierarchies have restrictions on label only members. See Table 123 on page 909.

In Figure 164 on page 921, the All Merchandise hierarchy and the High End Merchandise hierarchy are stored hierarchies. The All Merchandise member and the High End Merchandise member are the tops of the hierarchies and are both tagged as top of a stored hierarchy.

➤ To specify a stored hierarchy, tag the top member of the hierarchy as top of a stored hierarchy using a tool:

Tool	Topic	Location
Administration Services	Defining Hierarchies in Aggregate Storage Outlines	*Oracle Essbase Administration Services Online Help*
MaxL	**import database**	*Oracle Essbase Technical Reference*

The following members can be tagged as top of a stored hierarchy:

- **A dimension member (generation 1).** If a dimension member is tagged as top of a stored hierarchy, the entire dimension is considered a single stored hierarchy, and no other member in the dimension can be tagged as top of a stored hierarchy or top of a dynamic hierarchy.
- **The children of the dimension member (generation 2).** If a generation 2 member is tagged as top of a stored hierarchy, all generation 2 members in the dimension also must be tagged as either top of a stored hierarchy or top of a dynamic hierarchy. The first hierarchy in the dimension must be a stored hierarchy.

The dimension tagged as accounts is automatically considered a dynamic hierarchy. You cannot specify the accounts dimension as a stored hierarchy.

Dynamic Hierarchies

To evaluate a dynamic hierarchy, Essbase calculates, rather than aggregates, the members and formulas. The order in which members and formulas are evaluated is defined by the solve order property. See "Calculation Order" on page 973.

At the time of retrieval, Essbase calculates the required member combinations and calculates any required outline member formulas. Because dynamic hierarchies are calculated, the data retrieval time may be longer than for data retrieved from stored hierarchies. However, when you design your database, dynamic hierarchies provide the following advantages:

- They can contain any consolidation operator.
- They can have formulas.

➤ To specify a dynamic hierarchy, tag the top member of the hierarchy as top of a dynamic hierarchy using a tool:

Tool	Topic	Location
Administration Services	Defining Hierarchies in Aggregate Storage Outlines	*Oracle Essbase Administration Services Online Help*
MaxL	**import database**	*Oracle Essbase Technical Reference*

The following members can be tagged as top of a dynamic hierarchy:

- **A dimension member (generation 1).** If a dimension member is tagged as top of a dynamic hierarchy, the entire dimension is considered a single dynamic hierarchy, and no other member in the dimension can be tagged as top of a dynamic hierarchy or top of a stored hierarchy.
- **The children of the dimension member (generation 2).** If a generation 2 member is tagged as top of a dynamic hierarchy, all generation 2 members in the dimension must also be tagged as either top of a dynamic hierarchy or top of a stored hierarchy. The first hierarchy in the dimension must be a stored hierarchy.

Note:

If a member has the no-consolidation operator (~) on all its children, the member must be tagged label only.

The dimension tagged accounts is automatically considered a dynamic hierarchy. You cannot specify the accounts dimension as a stored hierarchy.

Essbase cannot select dynamic hierarchy members for an aggregate view. See "Aggregating an Aggregate Storage Database" on page 977.

Alternate Hierarchies

An alternate hierarchy may be modeled in either of the following ways:

- As an attribute dimension, which uses attributes to classify members logically within the dimension (for example, a Product dimension can have attributes such as Size and Flavor). See Chapter 10, "Working with Attributes."

  Note:

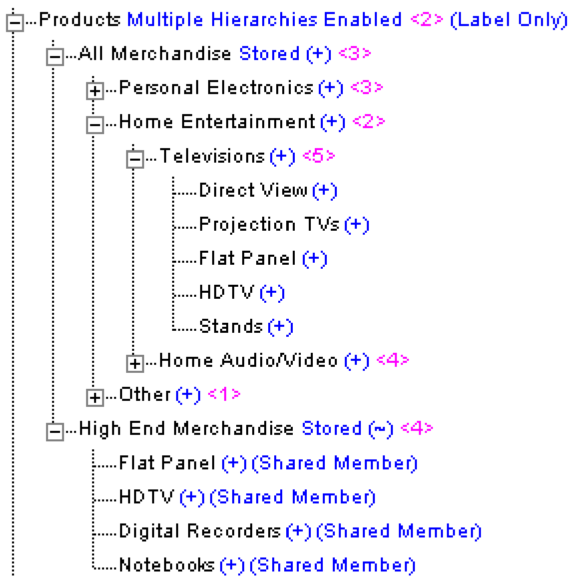
  If you use an attribute dimension to create an alternate hierarchy, you can create a crosstab report query of members in the attribute dimension with members in the base dimension. For example, a crosstab report of product sales information could show size attributes (such as small and large) as column headings and products as row headings. If you use shared members to create an alternate hierarchy, you cannot create an equivalent crosstab report query of the shared members with the nonshared members in the primary hierarchy.

- As a hierarchy of shared members. The alternate hierarchy has shared members that refer to nonshared members of previous hierarchies in the outline. The shared members roll up according to a different hierarchy from the nonshared members to which they refer. Shared members on dynamic hierarchies can have formulas. See "Understanding Shared Members" on page 149. Table 129 shows the hierarchies for the ASOsamp.Sample database. The Products dimension is shown in Figure 164 on page 921.

Table 129 Example Hierarchies and Alternate Hierarchies for the Product Dimension of ASOsamp.Sample

Product	Hierarchy	Alternate Hierarchy (containing shared members)
Flat Panel	Products, All Merchandise, Personal Electronics, Home Entertainment, Televisions	Products, High End Merchandise
HDTV	Products, All Merchandise, Personal Electronics, Home Entertainment, Televisions	Products, High End Merchandise

Figure 164 Aggregate Storage Outline Displaying the Alternate Hierarchy High End Merchandise on the Product Dimension of ASOSamp.Sample



```
Products Multiple Hierarchies Enabled <2> (Label Only)
    All Merchandise Stored (+) <3>
        Personal Electronics (+) <3>
        Home Entertainment (+) <2>
            Televisions (+) <5>
                Direct View (+)
                Projection TVs (+)
                Flat Panel (+)
                HDTV (+)
                Stands (+)
            Home Audio/Video (+) <4>
        Other (+) <1>
    High End Merchandise Stored (~) <4>
        Flat Panel (+) (Shared Member)
        HDTV (+) (Shared Member)
        Digital Recorders (+) (Shared Member)
        Notebooks (+) (Shared Member)
```

The following restrictions apply when creating alternate hierarchies in aggregate storage outlines:

- The nonshared instance of the member must occur in the outline before any shared instances of the member. For example, in Figure 164, the member HDTV occurs in the All Merchandise hierarchy before it occurs as a shared member in the alternate hierarchy of High End Merchandise.
- The first hierarchy in a dimension where multiple hierarchies are enabled cannot contain a shared member.
- Stored hierarchy dimensions cannot have shared members. Stored hierarchies within a multiple hierarchies dimension can have shared members.
- To ensure that values are not double-counted, a stored hierarchy cannot contain multiple copies of the same shared member. For example, a stored hierarchy cannot contain a shared

member and any of its ancestors. In Figure 164 on page 921, you cannot add the shared member "Televisions" as a child of "High End Merchandise," because doing so would make "Televisions" a sibling of its children, shared members "Flat Panel" and "HDTV," causing the values of "Flat Panel" and "HDTV" to be added twice.

- Nonshared instances of a member must be in the same dimension as the shared member (same for block storage outlines).
- A stored hierarchy cannot contain a nonshared instance and a shared instance of the same member.
- A stored hierarchy can contain a shared instance of a dynamic hierarchy member only if the dynamic hierarchy member is a level 0 member without a formula.

Note:

In an aggregate storage database, a shared member automatically shares any attributes that are associated with its nonshared member. This also applies to an implied shared member (for example, a member that has only one child). See "Understanding Implied Sharing" on page 152. You can prevent implied sharing by setting the Never Share property; see "Determining How Members Store Data Values" on page 147. This behavior of shared members and attributes in aggregate storage databases is different from the behavior in block storage databases.

Attribute Dimensions

This topic provides information on the differences between aggregate storage and block storage databases with regard to attribute dimensions. To use the information in this topic, you should be familiar with attribute dimension concepts for block storage databases. See Chapter 10, "Working with Attributes."

The following information applies to attribute dimensions when used on aggregate storage databases:

- Only the addition (+) consolidation operator is available for attribute dimensions.
- For a given attribute dimension, all associations must be with one level of the base dimension. For example, in the ASOSamp.Sample database, associations for the Store Manager attribute dimension are with level 0 of the Stores dimension. The following restrictions apply to attribute associations:
  - Level 0: You can associate attributes with any level 0 member of a dynamic or stored hierarchy that does not have a formula.
  - Non-level 0: You can associate attributes only to upper level members in the primary stored hierarchy.

Attribute dimensions do not have hierarchy types. You cannot specify an attribute dimension as a dynamic or stored hierarchy. Essbase treats attribute dimensions as stored alternate hierarchies of the base dimension. For example, in the ASOSamp.Sample database, Essbase treats the Store Manager attribute dimension as if the Store Manager dimension were a stored alternate hierarchy of the Stores dimension.

When using query tracking, Essbase considers queries on attribute dimension data and may include attribute dimension members in aggregate view selections. See "Selecting Views Based on Usage" on page 982 and "Calculating Aggregate Storage Databases" on page 971.

Note:

Queries on attribute members that are associated with non-level 0 members return values for descendants of the non-level 0 member. This behavior of queries on attribute members in aggregate storage databases is different from the behavior in block storage databases.

Design Considerations for Attribute Queries

When selecting and building views based on attribute query data, some queries on attribute data are always dynamically calculated at the time of retrieval, which may affect query performance.

Every query involving attribute dimension members must also include at least one member from the base dimension. If the query involves a single attribute dimension and a sum-of-all dimension member, Essbase aggregates the query data, potentially improving query performance. In other cases, Essbase must calculate the query at the time of retrieval.

The following table describes attribute query types and how Essbase calculates the query:

Table 130 Attribute Queries and Calculation Performance

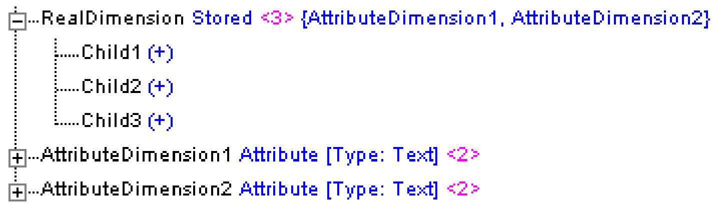
Attribute Query Type	Query Calculation Type
Query involves a sum-of-all base dimension member and members from one attribute dimension.	Essbase can aggregate query data, potentially improving query performance.
Query involves any member of the base dimension and members from multiple attribute dimensions.	Essbase calculates the query at the time of retrieval based on the level 0 input data.
Query involves any child member of the base dimension member (or dimension member that is tagged as label-only) and members from one attribute dimension.	Essbase calculates the query at the time of retrieval based on the level 0 input data, or on data from aggregations on the base dimension.

In the outline displayed in Figure 165 on page 924, RealDimension is the sum of all its descendents (it is not tagged as label-only). If a query involves one or more members from a single attribute dimension (for example, AttributeDimension1), crossed with the base dimension member (RealDimension), Essbase can build aggregate cells for the data, potentially improving query performance.

The following queries, however, are always calculated at the time of retrieval:

- Any query requesting data for members from an attribute dimension (for example AttributeDimension1), and any of the children of RealDimension is calculated dynamically at retrieval time based on the level 0 input data or on data from aggregations.
- Any query requesting data from multiple attribute dimensions (for example AttributeDimension1 and AttributeDimension2) and a base member dimension (for example RealDimension) is calculated dynamically at retrieval time based on level 0 input data.

Figure 165   Outline for Attribute Query Example



```
RealDimension Stored <3> {AttributeDimension1, AttributeDimension2}
    Child1 (+)
    Child2 (+)
    Child3 (+)
AttributeDimension1 Attribute [Type: Text] <2>
AttributeDimension2 Attribute [Type: Text] <2>
```

Design Considerations for Aggregate Storage Outlines

This topic lists the key design considerations when you create aggregate storage database outlines. For an example of implementing these design considerations, see the ASOSamp.Sample database. Consider the following information when designing an aggregate storage outline:

- Use stored hierarchies (rather than dynamic hierarchies) as much as possible.
- Use alternate hierarchies (shared members) only when necessary.
- Minimize the number of hierarchies. (For example, each additional stored hierarchy slows down view selection and potentially increases the size of the aggregated data).
- If a hierarchy is a small subset of the first hierarchy, consider making the small hierarchy a dynamic hierarchy. Considerations include how often the hierarchy data is queried and the query performance impact when it is dynamically queried at the time of retrieval.
- The performance of attributes is the same as for members on a stored hierarchy.
- The higher the association level of an attribute to the base member, the faster the retrieval query. (See also, "Design Considerations for Attribute Queries" on page 923).

Query Design Considerations for Aggregate Storage

When querying data from a dimension that has multiple hierarchies, query performance may improve if you query the data in the following way:

1. Select the hierarchy that you want to query.
2. Navigate to find the detailed data (for example, by zooming in on the hierarchy in Spreadsheet Toolkit).

Including dynamic hierarchy members and stored hierarchy members in the same query may require a large internal memory cache, which decreases query performance.

Understanding the Compression Dimension for Aggregate Storage Databases

By default, the compression dimension in an aggregate storage database is the Accounts dimension. Changing the compression dimension triggers a full restructure of the database. Essbase requires the compression dimension to be a single dynamic hierarchy. If the dimension has a different hierarchy setting, such as multiple hierarchies, it will be set to single dynamic hierarchy automatically. The original hierarchy setting is lost (setting a different dimension as compression does not return the original hierarchy setting). Attribute dimensions cannot be compression dimensions, nor can dimensions with attributes associated to them.

The choice of compression dimension can significantly affect performance. A good candidate for a compression dimension is one that optimizes data compression while maintaining retrieval performance. This topic provides information about choosing an optimal compression dimension.

Note:

The information in this topic applies to loaded databases. See "Loading Data into Aggregate Storage Databases" on page 957.

Maintaining Retrieval Performance

Because compression dimensions are dynamically calculated, you must take into account design considerations for dynamically calculated dimensions when choosing a compression dimension. Dynamic dimensions are calculated at the time of retrieval, so the data retrieval time is longer than for stored hierarchies.

If a dimension with a large number of level 0 members is tagged as compression, upper-level queries take longer because they require many level 0 members to be retrieved. If users will be doing many upper-level retrievals on a large dimension, it is not a good candidate for a compression dimension.

Managing Database Compression While Maintaining Retrieval Performance

Another consideration when choosing a compression dimension is how well it is expected to compress the database. The size of the compressed database changes depending on which dimension you tag as compression.

In Administration Services, you can view compression estimates and then choose a compression dimension in the same dialog box. Selecting a new compression dimension in Administration Services restructures the outline automatically.

➤ To view the expected level 0 size of the database for each dimension when hypothetically tagged as compression, and to choose a compression dimension, see "Selecting a Compression Dimension for Aggregate Storage" in the *Oracle Essbase Administration Services Online Help*.

Viewing Compression Estimation Statistics

In Administration Services and in MaxL, you can view detailed compression and query statistics. You can view the number of stored level 0 members, which affects retrieval performance; the average bundle fill and average value length, which affect compression; and the level 0 size.

➤ To view detailed query and compression statistics, use a tool:

Tool	Topic	Location
Administration Services	Database Properties Window	*Oracle Essbase Administration Services Online Help*
MaxL	Query Database (Aggregate Storage)	*Oracle Essbase Technical Reference*

The following sections describe each of the compression and query related statistics.

Stored level 0 members

Dimensions with a large number of stored level 0 members do not perform well if tagged Compression. As with any dynamically calculated dimension, upper-level retrievals from compression dimensions generally are slow. See "Maintaining Retrieval Performance" on page 925.

Average bundle fill

Compression is more effective if values are grouped together in consecutive members on dimensions or hierarchies rather than spread throughout the outline with lots of `#missing` data between values. Essbase saves memory by storing information about the location and contents of the groups rather than storing it separately for each of the members. The average bundle fill is the average number of values stored in the groups. It can vary between 1 and 16, with 16 being the best. Choosing a compression dimension that has a higher average bundle fill means that the database compresses better.

In some outlines, you can improve compression by ordering the numbers in the compression dimension so that members that are frequently populated are grouped together. When populated members are grouped together, more values fit into each bundle, increasing the average bundle fill and improving compression.

Average value length

The average value length is the average storage size, in bytes, required for the stored values in the cells. It can vary between 2 bytes and 8 bytes with 2 bytes being the best. Without compression, it takes 8 bytes to store a value in a cell. With compression, it can take fewer bytes, depending on the value length. For example, 10.050001 might take 8 bytes to store even when compressed, but 10.05 may only take 2 bytes–4 bytes to store when compressed. Dimensions with a smaller average value length compress the database better.

Rounding the data values to no more that two digits after the decimal point can reduce the average value length, improving compression.

Expected level 0 size

This field indicates the estimated size of the compressed database. A smaller expected level 0 size indicates that choosing this dimension is expected to enable better compression.

Verifying Outlines

Aggregate storage outline files have the same file extension (`.otl`) as block storage database outline files and are stored in an equivalent directory structure. When you save an outline, Essbase verifies it for errors. You can also verify the accuracy of an outline before you save it. Some block storage database features do not apply to aggregate storage databases, and the verification process considers the rules for aggregate storage databases. See Chapter 57, "Comparison of Aggregate and Block Storage."

➤ To verify an aggregate storage outline, see "Verifying Outlines" in the *Oracle Essbase Administration Services Online Help*.

Outline Paging

Aggregate storage database outlines are pageable. This feature may significantly reduce memory usage for very large database outlines. For aggregate storage databases, Essbase preloads part of the database outline into memory. Then, during data retrieval, Essbase pages other parts of the outline into memory as required.

When you create an aggregate storage database, the outline is created in a pageable format. When you use the Aggregate Storage Outline Conversion Wizard to convert an existing block storage outline to aggregate storage, the outline is automatically converted to a pageable format.

Paging an outline into memory enables Essbase to handle very large outlines (for example, 10 million or more members) but potentially increases data retrieval time. You can customize outline paging to obtain the optimum balance between memory usage and data retrieval time. See "Optimizing Outline Paging" on page 929.

Note:

Aggregate storage databases that have pageable outlines contain memory pages, and therefore their outline files may be larger than binary block storage database outline files.

Outline Paging Limits

The maximum size of a buildable outline (the number of members) depends on several factors:

- The available memory for Essbase
- The amount of memory in Essbase allocated for other uses
- The amount of memory required for each member (and aliases for each member)

Table 131 shows the amount of addressable memory available for Essbase for different operating systems.

Table 131 Addressable Memory Per Operating System

Operating System	Addressable Memory
Windows 2000, Windows 2003	2 GB addressable memory 1.85 GB available to any application, including Essbase
Windows 2000 Advanced Server, Windows 2003 Advanced Server	3 GB Requires a setting in the `boot.ini` file
AIX 5.*x*	3.25 GB Up to 13 (256 MB) segments. Requires setting the LDR_CNTRL environment variable to: `0xD0000000@DSA`
HP-UX	2.9 GB Requires using the following command to set the addressable memory for the Essbase server process, ESSSVR: `chatr +q3p enable ESSSVR`
Solaris, Linux	3.9 GB available by default

Essbase uses about 40 MB of memory on startup. In addition, the various caches require the following memory allocations:

- Outline paging cache: 8 MB
- Aggregate storage data cache: 32 MB
- Aggregate storage aggregation cache: 10 MB

Therefore, the initial memory footprint for Essbase is about 90 MB. In addition, memory must be allocated to process incoming query requests. Typical memory to reserve for this purpose is about 300 MB. The total memory allocated for Essbase is therefore 390 MB.

On a Windows system with 1.85 GB of addressable memory, the amount available to build and load the outline is about 1.46 GB (1.85 GB - 390 MB = 1.46 GB).

The maximum outline size depends on whether it is built using a dimension build or from an outline already loaded into Essbase.

Dimension Build Limit

To build the outline by using a dimension build, Essbase allocates about 100 bytes per member, plus the size of the member name, plus the size of all alias names for the member (up to 10 aliases are allowed).

For a sample outline (using a single byte codepage) where the average member name is 15 characters and there is one alias (of 20 characters) per member, the memory requirement for each member that is added:

100 + 15 + 20 bytes = 135 bytes

The total number of members that can be added in a dimension build is the available memory (1.46 GB, or 153,092,060 bytes) divided by the number of bytes per member (135), approximately 11 million members.

On systems with more than 2 GB of addressable memory, the outline can be larger in proportion to the extra memory that is available.

When the dimension build is complete, a *databaseName*.otn file is saved in the database directory. The .otn file is used as input for the outline restructuring process, which replaces the old outline with the new one. During restructuring, two copies of the outline are loaded into memory, the old one (potentially empty), and the new one, so the maximum size of an outline that can be restructured depends on the size of the old outline.

In a dimension build, which starts with an empty outline, only one outline is loaded into memory.

Loaded Outline Limit

The memory requirement for an outline loaded into Essbase at runtime or during restructuring is different from the memory requirements for a dimension build. Essbase allocates about 60 bytes per member, plus the size of the member name plus 5 bytes, plus the size of all alias names for the member (up to 10 aliases are allowed) plus 5 bytes. For a sample outline where the average member name is 15 characters and there is one alias (of 20 characters) per member, the memory requirement for each member that is added:

60 + 15 + 5 + 20 + 5 bytes = 105 bytes per member

Assuming 1.46 GB of available memory, the maximum size of an outline that can be loaded is one with 14 million members (1.46 GB / 105 bytes).

The 14 million members are the sum of two outlines that are loaded during restructuring. For example, if an existing outline has 5 million members, the new outline can have a maximum of 9 million members. In an incremental dimension build, it is recommended to build the smaller dimensions first and the larger ones last to allow for a maximum outline size.

Optimizing Outline Paging

Depending on how you want to balance memory usage and data retrieval time, you can customize outline paging for aggregate storage outlines by using one or more of the following settings in the essbase.cfg file:

- PRELOADMEMBERNAMESPACE to turn off preloading of the member namespace.
- PRELOADALIASNAMESPACE to turn off preloading of the alias namespace.

See the *Oracle Essbase Technical Reference*.

When Essbase loads an outline, it attempts to load into memory the namespaces for both member names and all alias tables to allow optimal performance during name lookup. Name lookup is used during data load, and during report, spreadsheet, and MDX queries.

If the available memory does not allow all namespaces to be preloaded, the alias namespace is left on disk and paged in on demand. If there is still not enough memory, the member namespace is also left on disk and paged in on demand.

If memory calculations indicate that it is possible to build and restructure a particular outline if one or both namespaces are not preloaded, you can set one or both of the PRELOADMEMBERNAMESPACE and PRELOADALIASNAMESPACE configuration settings to FALSE to turn off preloading the namespaces.

Not preloading the namespaces will have a severe performance impact but could be a temporary measure to build a very large outline. After the outline is built and restructured Essbase can be restarted with the namespace settings set back to their default TRUE (on) setting.

Compacting the Outline File

When you delete members from an aggregate storage outline, the corresponding records of members in the outline file (`.otl` file) are marked as deleted but remain in the file. The outline file continues to grow as members are deleted and added. You can minimize the outline file size by compacting the file to remove the records of deleted members. Compacting the outline file causes Essbase to restructure the outline and can take place only when no other users or processes are actively using the database. Compacting the outline does not cause Essbase to clear the data.

➤ To compact an outline file, use a tool:

Tool	Topic	Location
Administration Services	Compacting the Outline File	*Oracle Essbase Administration Services Online Help*
MaxL	**alter database**	*Oracle Essbase Technical Reference*
ESSCMD	COMPACTOUTLINE	*Oracle Essbase Technical Reference*

Developing Formulas on Aggregate Storage Outlines

Formulas calculate relationships between members in a database outline. If you are familiar with using formulas on block storage outlines, consider the following differences when using formulas on aggregate storage outlines:

- Essbase provides a native calculation language (the Calc language, or Calc) to write formulas on block storage outlines. To write formulas for aggregate storage outlines, the MDX (Multidimensional Expressions) language is required.
- Apply formulas directly to members in the database outline. For block storage databases, formulas can be placed in a calculation script. For aggregate storage databases, you cannot place formulas in a calculation script.

The current chapter concentrates on using MDX to write formulas on aggregate storage databases. For information about using MDX to write queries, see Chapter 34, “Writing MDX Queries.” For information about writing formulas for block storage outlines, see Chapter 22, “Developing Formulas.” Also see the MDX section of the *Oracle Essbase Technical Reference*.

Using MDX Formulas

An MDX formula must always be an MDX numeric value expression. In MDX, a numeric value expression is any combination of functions, operators, and member names that does one of the following actions:

- Calculates a value
- Tests for a condition
- Performs a mathematical operation

A numeric value expression is different from a set expression. A set expression is used on query axes and describes members and member combinations. A numeric value expression specifies a value.

A numeric value expression is used in queries to build calculated members, which are logical members created for analytical purposes in the WITH section of the query, but which do not exist in the outline.

The following query defines a calculated member and uses a numeric value expression to provide a value for it:

```
WITH MEMBER
 [Measures].[Prod Count]
AS
 'Count (
    Crossjoin (
     {[Units]},
     {[Products].children}
    )
  )', SOLVE_ORDER=1
SELECT
 {[Geography].children}
ON COLUMNS,
 {
  Crossjoin (
     {[Units]},
     {[Products].children}
    ),
   ([Measures].[Prod Count], [Products])
 }
ON ROWS
FROM
 ASOSamp.Sample
```

In the sample query, the WITH clause defines a calculated member, Product Count, in the Measures dimension, as follows:

```
WITH MEMBER
 [Measures].[Prod Count]
```

The numeric value expression follows the WITH clause and is enclosed in single quotation marks. In the sample query, the numeric value expression is specified as follows:

```
'Count (
    Crossjoin (
      {[Units]},
      {[Products].children}
    )
  )'
```

The SOLVE_ORDER property specifies the order in which members and formulas are evaluated. See "Calculation Order" on page 973.

Note:

For an explanation of the syntax rules used to build the numeric value expression in the example, see the documentation in the *Oracle Essbase Technical Reference* for the Count, CrossJoin, and Children functions.

A numeric value expression also can be used as an MDX formula to calculate the value of an existing outline member.

Therefore, rather than creating the example query, you can create an outline member on the Measures dimension called Prod Count that is calculated in the outline in the same way that the hypothetical Prod Count was calculated in the sample query.

➤ To create a calculated member with a formula:

1 **Create a member.**

2 **Attach an MDX formula to the member.**

Assuming that you created the example Prod Count member, you would use the following formula, which is the equivalent of the numeric value expression used to create the calculated member in the example query:

```
Count(Crossjoin ( {[Units]}, {[Products].children}))
```

3 **Verify the formula by verifying the outline.**

When you retrieve data from the aggregate storage database, the formula is used to calculate the member value.

You can use substitution variables within formulas. For example, you could define a substitution variable named "EstimatedPercent" and provide different percentages as substitution variable values. See "Using Substitution Variables" on page 120.

Before applying formulas to members in the outline, you can write MDX queries that contain calculated members. When you can write an MDX query that returns the calculated member results that you want, you are ready to apply the logic of the numeric value expression to an outline member and validate and test the expression. For information about writing MDX queries, see Chapter 34, "Writing MDX Queries". For syntax information about MDX, see the *Oracle Essbase Technical Reference*.

Formula Calculation for Aggregate Storage Databases

Essbase calculates formulas in aggregate storage outlines only when data is retrieved. Calculation order may affect calculation results. Whenever you use MDX formulas on multiple dimensions in an aggregate storage outline, it is good practice to set the solve order for each member or dimension. See "Calculation Order" on page 973.

Note:

When designing an aggregate storage database calculation, consider that aggregate storage database members with MDX formulas are dynamically calculated. The dynamically calculated members have a value of #MISSING until they are queried.

Formula Syntax for Aggregate Storage Databases

When you create member formulas for aggregate storage outlines, observe the following rules:

- Enclose member names in brackets (`[]`) if they meet any of the following conditions:
  - Start with a number or contains spaces; for example, `[100]`. Brackets are recommended for all member names, for clarity and code readability.
  - Are the same as an operator or function name. See the *Oracle Essbase Technical Reference* for a list of operators and functions.
  - Include a nonalphanumeric character; for example, a hyphen (`-`), an asterisk (`*`), or a slash (`/`).
- Use the IIF function to write conditional tests with a single else condition. The syntax for the IIF function does not require an ELSEIF keyword to identify the else condition nor an ENDIF keyword to terminate the statement. You can nest IIF functions to create a more complex formula.
- Use the CASE, WHEN, THEN construct to write conditional tests with multiple conditions.
- Be certain that tuples are specified correctly. A tuple is a collection of members with the restriction that no two members can be from the same dimension. Enclose tuples in parentheses; for example, `(Actual, Sales)`.
- Be certain that sets are specified correctly. A set is an ordered collection of one or more tuples. When a set has more than one tuple, the following rule applies: In each tuple of the set, members must represent the same dimensions as do the members of other tuples of the set. Additionally, the dimensions must be represented in the same order. In other words, all tuples of the set must have the same *dimensionality*.

  See "Rules for Specifying Sets" on page 554.

  Enclose sets in braces, for example:

  ```
  { [Year].[Qtr1], [Year].[Qtr2], [Year].[Qtr3], [Year].[Qtr4] }
  ```

Creating Formulas on Aggregate Storage Outlines

You use Formula Editor to create formulas. Formula Editor is a tab in the Member Properties dialog box in Outline Editor. Formulas are plain text. You can type the formulas directly into the formula text area, or you can create a formula in the text editor of your choice and paste it into Formula Editor.

You can also include formulas in a dimension build data source. See "Setting Field Type Information" on page 277.

➤ To create a formula, see "Creating Formulas for Aggregate Storage Databases" in the *Oracle Essbase Administration Services Online Help*.

Checking Formula Syntax

Essbase includes MDX-based syntax checking that tells you about syntax errors in formulas. For example, Essbase tells you if you have mistyped a function name or specified a nonexistent member. Unknown names can be validated against a list of function names. If you are not connected to Essbase Server or to the application associated with the outline, Essbase may connect you to validate unknown names.

Syntax checking occurs when you save a formula. Errors are displayed in the Messages panel. If an error occurs, you choose to save or not save the formula. If you save a formula with errors, you are warned when you verify or save the outline. When you calculate a formula with errors, the formula is ignored and the member is given a value of $MISSING.

A syntax checker cannot warn you of semantic errors in a formula. Semantic errors occur when a formula does not work as you expect. One way to find semantic errors in a formula is to place the numeric value expression that defines the formula into a query and run the query to verify that the results are as you expect. See "Using MDX Formulas" on page 931 to see how to place a formula into a query.

You can use MDX Script Editor to create a query. MDX Script editor provides features such as color coding and autocompletion of MDX syntax. See "About MDX Script Editor" in the *Oracle Essbase Administration Services Online Help*.

Displaying Formulas

➤ To display a formula, use tool:

Tool	Topic	Location
Administration Services	Creating Formulas for Aggregate Storage Databases	*Oracle Essbase Administration Services Online Help*
MaxL	**query database**	*Oracle Essbase Technical Reference*
ESSCMD	GETMBRCALC	*Oracle Essbase Technical Reference*

Composing Formulas on Aggregate Storage Outlines

The following sections discuss and give examples of how to write a variety of formulas for members in aggregate storage outlines.

Basic Equations for Aggregate Storage Outlines

You can apply a mathematical operation to a formula to create a basic equation. For example, the following formula is applied to the Avg Units/Transaction member in the ASOSamp.Sample database:

```
[Units]/[Transactions]
```

The formula in Avg Units/Transaction divides the number of units by the number of transactions to arrive at the average number of units per transaction.

In aggregate storage outlines, members cannot be tagged as expense items. Therefore, functions in Calc, such as @VAR and @VARPER, which determine the difference between two members by considering expense tags, are not relevant in aggregate storage outlines.

The MDX subtraction operator can be used to calculate the difference between two members. For example, the following formula can be applied to a new member, called Price Diff, in ASOSamp.Sample to calculate the difference between the price paid and the original price:

```
[Price Paid]-[Original Price]
```

Members Across Dimensions in Aggregate Storage Outlines

ASOSamp.Sample provides a formula on a member called % of Total. This member formula identifies the percentage of the Measures total that is produced by Transactions. The formula for % of Total:

```
Transactions/(Transactions,Years,Months,
[Transaction Type],[Payment Type],Promotions,Age,
[Income Level],Products,Stores,Geography)
```

The formula specifies a member (Transactions) divided by a tuple (Transactions, Years, ...). The formula lists a top member from every dimension to account for all Transaction data in the cube; that is, not Transaction data for the Curr Year member but Transaction data for all members of the Years dimension, not Transaction data for months in the first two quarters but Transaction for all months, and so on. In this way, the value of % of Total represents the percentage of the Measures total that are produced by Transactions.

Conditional Tests in Formulas for Aggregate Storage Outlines

You can define a formula that uses a conditional test or a series of conditional tests to determine the value for a member. Use the IIF function to perform a test with one else condition. You can nest IIF functions to create a more complex query.

The example specifies a formula for a member that represents the price the company must pay for credit card transactions, which includes a 5% charge. The following example assumes that the Credit Price member has been added to the Measures dimension of the ASOSamp.Sample database. Credit Price has the following formula, which adds 5% to Price Paid when the payment type is a credit card.

```
IIF (
     [Payment Type].CurrentMember=[Credit Card],
     [Price Paid] * 1.05, [Price Paid]
)
```

Use the CASE, WHEN, THEN construct to create formulas with multiple tests and else conditions.

The Filter function returns the tuples of the input set that meet the criteria of the specified search condition. For example, to establish a baseline (100) for all products, you can add a Baseline member and create a formula for it, as follows:

```
Count(Filter(Descendants([Personal
Electronics],[Products].Levels(0)),[Qtr1] > 100.00))
```

Specifying UDAs in Formulas in Aggregate Storage Outlines

UDAs are words or phrases that you create for a member. For example, in Sample.Basic, top-level members of the Market dimension have the UDA Small Market or the UDA Major Market.

The Major Market example used in this topic shows how to create a formula for a member that shows the sum of sales for all major market members. The example assumes that a new member (Major Market Total) has been added to Sample.Basic.

1. MDX provides a Boolean function, IsUDA, which Returns TRUE if a member has the associated UDA tag. The following syntax returns TRUE if the current member of the Market dimension has the UDA “Major Market”:

   ```
   IsUda([Market].CurrentMember, "Major Market")
   ```

2. A Filter function, when used with IsUDA (as shown in the following syntax), cycles through each member of the Market dimension and returns a value for each member that has the Major Market UDA:

   ```
   Filter([Market].Members, IsUda([Market].CurrentMember, "Major
   Market"))
   ```

3. The Sum function adds the values returned by the Filter function; for the Major Market example, the following formula is produced:

   ```
   Sum (Filter([Market].Members, IsUda([Market].CurrentMember, "Major
   Market")))
   ```

   This formula is attached to the Major Market Total member.

Using a Transparent Partition to Enable Write-Back for Aggregate Storage Databases

With a write-back partition, you can update data on-the-fly in the target block storage database while the data in the source aggregate storage database remains unchanged. Creating a write-back partition potentially decreases calculation time and reduces database size.

When creating write-back partitions, follow these guidelines:

- Create the block storage database in a separate application from the one in which the aggregate storage database is located.

  Typically, the block storage database contains a subset of the dimensions in the aggregate storage database.
- Create a transparent partition based on where you want the data to be stored. Make the block storage database the target and the aggregate storage database the source.

  See Chapter 14, "Designing Partitioned Applications."

  Note:

  You may want to partition on the time dimension if data for some time periods is stored in the aggregate storage database and data for other time periods is stored in the block storage database. For example, if you have actual data for January through March, which is stored in an aggregate storage database, and you want to budget for the last nine months of the year using write-back members in a block storage database.

Users query and write-back to the block storage database. Queries are processed by the block storage database or transparently by the aggregate storage database.

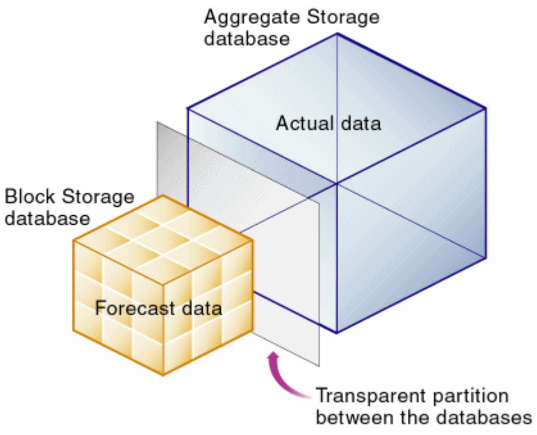
➤ To create an aggregate storage and block storage write-back partition, use a tool:

Tool	Topic	Location
Administration Services	Aggregate Storage Partition Wizard Creating Partitions	*Oracle Essbase Administration Services Online Help*
MaxL	**create database** **create partition**	*Oracle Essbase Technical Reference*

You need Database Manager permissions to create a partitioned application.

Figure 166 illustrates using a transparent partition for analyzing the variance between forecast and actual data:

Figure 166 Conceptual Diagram Showing a Transparent Partition Used for Analyzing Variance Between Forecast and Actual Data


Aggregate Storage database
Actual data
Block Storage database
Forecast data
Transparent partition between the databases

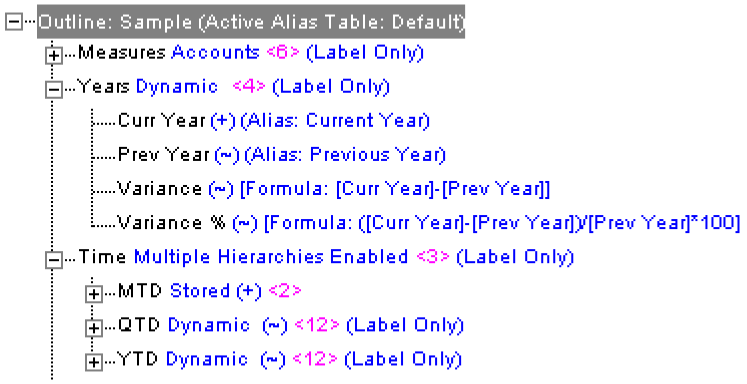
The following procedure is based on the aggregate storage sample database (ASOsamp.Sample), and uses the Administration Services Aggregate Storage Partition Wizard (see the *Oracle Essbase Administration Services Online Help*).

➤ To create a write-back partition:

1 **Select the ASOsamp.Sample database, which contains the actual data for the current year and for previous years.**

See Figure 167.

Figure 167 ASOSamp.Sample Aggregate Storage Database Outline



```
Outline: Sample (Active Alias Table: Default)
  Measures Accounts <6> (Label Only)
  Years Dynamic <4> (Label Only)
    Curr Year (+) (Alias: Current Year)
    Prev Year (~) (Alias: Previous Year)
    Variance (~) [Formula: [Curr Year]-[Prev Year]]
    Variance % (~) [Formula: ([Curr Year]-[Prev Year])/[Prev Year]*100]
  Time Multiple Hierarchies Enabled <3> (Label Only)
    MTD Stored (+) <2>
    QTD Dynamic (~) <12> (Label Only)
    YTD Dynamic (~) <12> (Label Only)
```

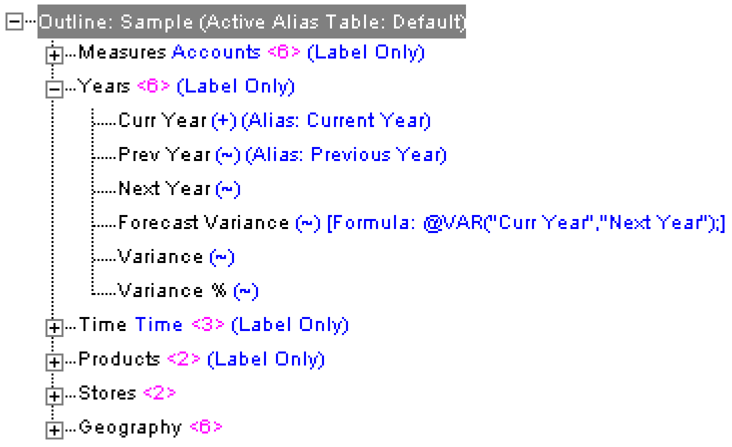
2 **Create a block storage database containing the following subset of dimensions from ASOsamp.Sample: Measures, Years, Time, Products, Stores, and Geography.**

See Figure 168.

3 **Edit the Years dimension to add the following members to the block storage database outline:**

- A member called Next Year, which will contain the forecast data.
- A member called Forecast Variance. Add a formula to this member to calculate the variance between actual and forecast data.

Figure 168 Block Storage Database Outline Showing Years Dimension Members for Calculating Variance Between Actual and Forecast Data



```
Outline: Sample (Active Alias Table: Default)
  Measures Accounts <6> (Label Only)
  Years <6> (Label Only)
    Curr Year (+) (Alias: Current Year)
    Prev Year (~) (Alias: Previous Year)
    Next Year (~)
    Forecast Variance (~) [Formula: @VAR("Curr Year","Next Year");]
    Variance (~)
    Variance % (~)
  Time Time <3> (Label Only)
  Products <2> (Label Only)
  Stores <2>
  Geography <6>
```

4 **Delete the following member formulas:**

- Measures dimension member formulas on Avg Units/Transaction and % of Total
- Years dimension member formulas on Variance and Variance%
- Time dimension member formulas under QTD and YTD

These formulas are expressions written in MDX that are copied from the aggregate storage database. MDX formula expressions cannot be interpreted in block storage databases.

5 **Link the databases with a transparent partition on the Years dimension, with the block storage database (forecast data) as the target and the aggregate storage database (actual data) as the source.**

Do not include the write-back members (Forecast and Variance) in the partitioned area.

Note:

When using the Administration Services Aggregate Storage Partition wizard, this step is automatic. The databases are automatically partitioned on the Years dimension because you selected the Years dimension in step 3. The write-back members are not included in the partitioned area.

You input forecast values into the block storage database write-back members. Because the added members are outside the partitioned area, you can write to them and then calculate data and generate reports based on updated data. The transparent partition provides a seamless view of both databases.

59 Aggregate Storage Time-Based Analysis

In This Chapter

The information in this chapter applies only to aggregate storage databases and is not relevant to block storage databases. Also see Chapter 57, “Comparison of Aggregate and Block Storage.”

Introduction

Most data analysis includes a time (history) perspective. Aggregate storage applications provide built-in time-based functionality through time dimensions and date-time dimensions.

Aggregate storage time dimensions are similar to block storage time dimensions. They enable tagging the accounts dimension for such time balance calculations as first, last, average, and to-date values. Time dimensions are easily modifiable.

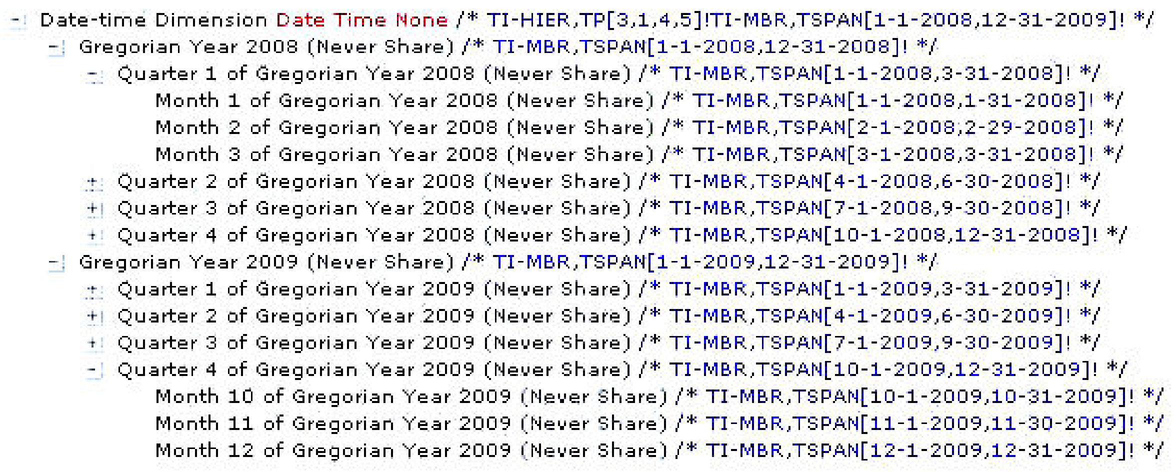
Aggregate storage date-time dimensions enable the first, last, and average time balance support. In addition, Administration Services provides the powerful Create Date-Time Dimension wizard for quick date-time dimension setup based on special calendars, plus linked attribute dimension setup enabling crosstab reporting of time-related data. Date-time dimensions have a rigid structure and are not easily modified once they are created.

The remainder of this chapter discusses date-time dimensions and the linked attribute dimensions that can be associated with them. For information about time balance calculations in aggregate storage outlines, see “Performing Time Balance and Flow Metrics Calculations in Aggregate Storage Accounts Dimensions” on page 987.

Understanding Date-Time Dimensions

Date-time dimensions contain a date hierarchy based on the selection of one of several standard corporate calendars. Dimension and member properties contain text strings that Essbase Server uses to manage the hierarchy, as shown in Figure 169.

Figure 169 Sample Date-Time Hierarchy



```
Date-time Dimension Date Time None /* TI-HIER,TP[3,1,4,5]!TI-MBR,TSPAN[1-1-2008,12-31-2009]! */
  Gregorian Year 2008 (Never Share) /* TI-MBR,TSPAN[1-1-2008,12-31-2008]! */
    Quarter 1 of Gregorian Year 2008 (Never Share) /* TI-MBR,TSPAN[1-1-2008,3-31-2008]! */
      Month 1 of Gregorian Year 2008 (Never Share) /* TI-MBR,TSPAN[1-1-2008,1-31-2008]! */
      Month 2 of Gregorian Year 2008 (Never Share) /* TI-MBR,TSPAN[2-1-2008,2-29-2008]! */
      Month 3 of Gregorian Year 2008 (Never Share) /* TI-MBR,TSPAN[3-1-2008,3-31-2008]! */
    Quarter 2 of Gregorian Year 2008 (Never Share) /* TI-MBR,TSPAN[4-1-2008,6-30-2008]! */
    Quarter 3 of Gregorian Year 2008 (Never Share) /* TI-MBR,TSPAN[7-1-2008,9-30-2008]! */
    Quarter 4 of Gregorian Year 2008 (Never Share) /* TI-MBR,TSPAN[10-1-2008,12-31-2008]! */
  Gregorian Year 2009 (Never Share) /* TI-MBR,TSPAN[1-1-2009,12-31-2009]! */
    Quarter 1 of Gregorian Year 2009 (Never Share) /* TI-MBR,TSPAN[1-1-2009,3-31-2009]! */
    Quarter 2 of Gregorian Year 2009 (Never Share) /* TI-MBR,TSPAN[4-1-2009,6-30-2009]! */
    Quarter 3 of Gregorian Year 2009 (Never Share) /* TI-MBR,TSPAN[7-1-2009,9-30-2009]! */
    Quarter 4 of Gregorian Year 2009 (Never Share) /* TI-MBR,TSPAN[10-1-2009,12-31-2009]! */
      Month 10 of Gregorian Year 2009 (Never Share) /* TI-MBR,TSPAN[10-1-2009,10-31-2009]! */
      Month 11 of Gregorian Year 2009 (Never Share) /* TI-MBR,TSPAN[11-1-2009,11-30-2009]! */
      Month 12 of Gregorian Year 2009 (Never Share) /* TI-MBR,TSPAN[12-1-2009,12-31-2009]! */
```

The dimension member at the top of the hierarchy shows the comment:

```
/* TI-HIER,TP[3,1,4,5]!TI-MBR,TSPAN[1-1-2008,12-31-2009]! */.
```

The TI-MBR,TSPAN[*date1*, *date2*] information enclosed between exclamation points (!) defines the time range covered by the dimension; in this case two years, Jan. 1, 2008–Dec. 31, 2009. Positioned in front of the time span, the array TI-HIER,TP[3,1,4,5] describes the hierarchy. The first number in the array, 3, indicates that the hierarchy contains three time periods. The remaining numbers in the array indicate what time periods are included, based on the following possible levels:

- 1—Year
- 2—Semester
- 3—Trimester
- 4—Quarter
- 5—Month
- 6—Period
- 7—Week
- 8—Day

Array TI-HIER,TP[3,1,4,5] indicates that the array contains three time periods: year, quarter, and month.

Using the same time span TI-MBR,TSPAN syntax, each member comments field defines the time range that it covers; for example, member “Quarter 1 of Gregorian Year 2008” covers Jan.

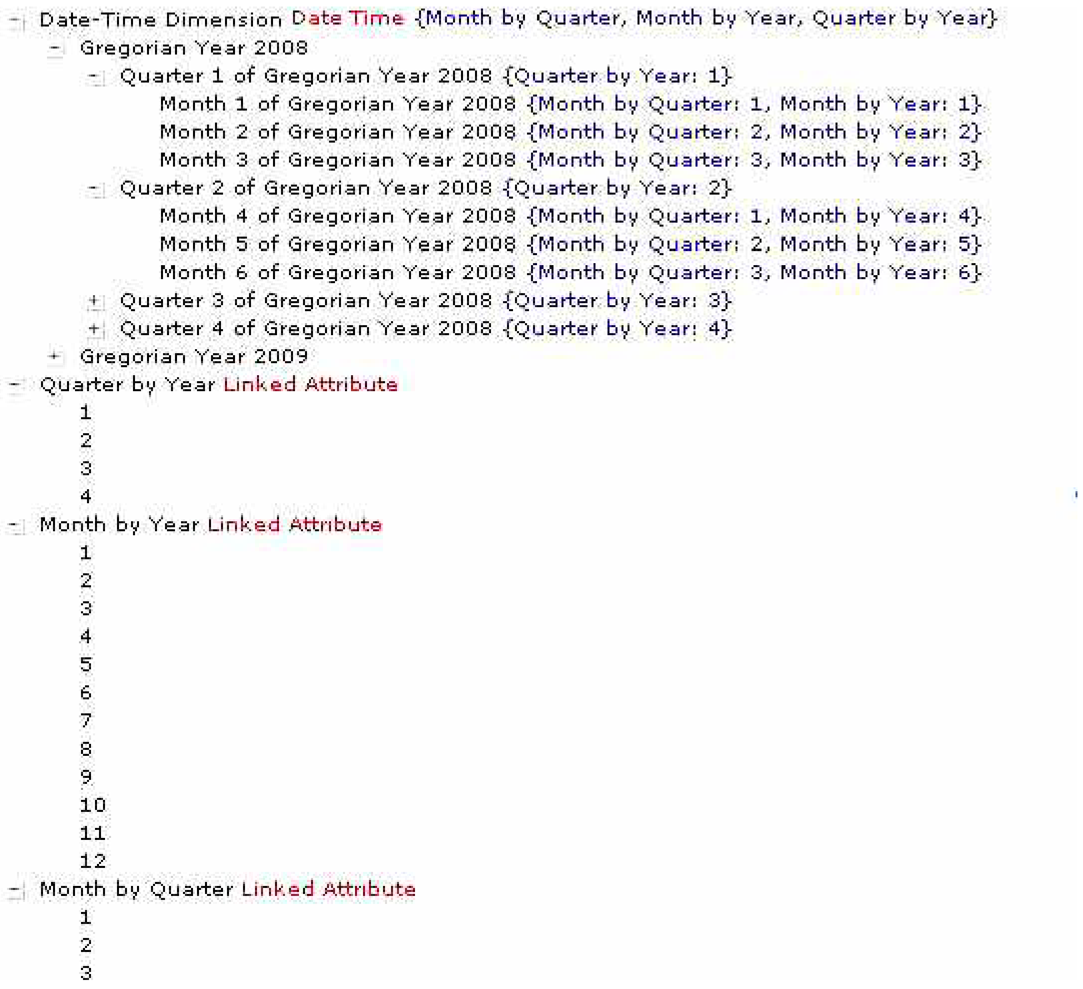
1, 2008–Mar. 31, 2008, and member “Quarter 2 of Gregorian Year 2008” covers Apr. 1, 2008–Jun. 30, 2008.

Understanding Linked Attributes

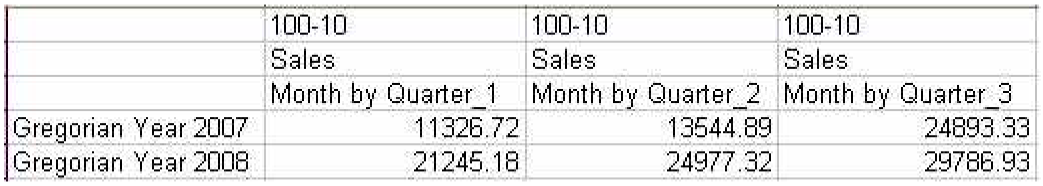
Analyzing data in terms of relative position within time periods can be useful; for example, knowing that some costs increase in the last week of each quarter can help you balance expenditures. Linked attributes enable data analysis based on relative time periods such as the first month of a quarter, or the fourth week of a month.

Linked attribute dimensions can be associated only with the date-time dimension. Defining linked attribute dimensions is part of the Create Date-Time Dimensions wizard. The linked attribute dimensions that you create are based on the time depths you select when you define the calendar for the date-time dimension. For example, as shown in Figure 170, if you select year, quarter, and month time depths and create linked attributes, the wizard creates three linked attribute dimensions (default dimension names): “Quarter by Year,” “Month by Year,” and “Month by Quarter.” Each attribute dimension contains members with sequential numeric names, indicating a relative position that will be used in analyzing the members associated with that attribute.

Figure 170   Sample Date-Time Dimension with Linked Attributes


```
− Date-Time Dimension Date Time {Month by Quarter, Month by Year, Quarter by Year}
    − Gregorian Year 2008
        − Quarter 1 of Gregorian Year 2008 {Quarter by Year: 1}
              Month 1 of Gregorian Year 2008 {Month by Quarter: 1, Month by Year: 1}
              Month 2 of Gregorian Year 2008 {Month by Quarter: 2, Month by Year: 2}
              Month 3 of Gregorian Year 2008 {Month by Quarter: 3, Month by Year: 3}
        − Quarter 2 of Gregorian Year 2008 {Quarter by Year: 2}
              Month 4 of Gregorian Year 2008 {Month by Quarter: 1, Month by Year: 4}
              Month 5 of Gregorian Year 2008 {Month by Quarter: 2, Month by Year: 5}
              Month 6 of Gregorian Year 2008 {Month by Quarter: 3, Month by Year: 6}
        + Quarter 3 of Gregorian Year 2008 {Quarter by Year: 3}
        + Quarter 4 of Gregorian Year 2008 {Quarter by Year: 4}
    + Gregorian Year 2009
− Quarter by Year Linked Attribute
      1
      2
      3
      4
− Month by Year Linked Attribute
      1
      2
      3
      4
      5
      6
      7
      8
      9
      10
      11
      12
− Month by Quarter Linked Attribute
      1
      2
      3
```

Linked attributes enable you to compare sales of the first month of a quarter against sales of the second and third months of a quarter in a simple crosstab analysis, as shown:



	100-10	100-10	100-10
	Sales	Sales	Sales
	Month by Quarter_1	Month by Quarter_2	Month by Quarter_3
Gregorian Year 2007	11326.72	13544.89	24893.33
Gregorian Year 2008	21245.18	24977.32	29786.93

Note:

Linked attributes cannot be assigned to year time-depth members.

Linked attribute dimensions reflect two time components, which are easier to understand if we use the default names; for example, “Quarter by Year.” Members of the “Quarter by Year” dimension are associated with quarter time-depth members such as “Quarter 1 of Gregorian Year 2008” in Figure 170 on page 944. The first time component, “Quarter,” is called the association level, because it indicates the date-time dimension members to which the attributes are associated.

The second time component in this example, “Year,” is called the attachment level. The attachment level indicates a member level in the associated date-time dimension. Each member of the linked attribute dimension contains a number that indicates a relationship between the association level and the attachment level. “Quarter by Year: 1” is the first attribute member “Quarter by Year: 2” is the second member, and so on.

The date-time dimension members associated with linked attribute “Quarter by Year: 1” are the first quarters of their respective years. All date-time dimension members with the attribute “Month by Quarter: 1” are the first months of their respective quarters. Thus linked attributes enable analysis of information based on a common periodic relationship such as first, second, and third.

Designing and Creating an Outline for Date-Time Analysis

The Administration Services Create Date-Time Dimension wizard uses your option selections to design a time-analysis structure including date-time dimension through calendar templates. You choose a calendar type, customize it to your business requirements, and select other options to fit your time-analysis needs. The wizard creates the following outline components:

- The date-time dimension.

  See “Understanding Date-Time Dimensions” on page 942.
- (Optional) Linked attribute dimensions and member associations.

  See “Understanding Linked Attributes” on page 943.
- (Optional) Attribute dimensions associated with day-level members. These are standard Boolean or text attributes based on specific dates or days of the week. You can use these attributes to include or exclude day-level information. For example, the wizard enables defining the following types of situations:
  - Assigning a Boolean attribute dimension such as “Weekend” to selected days of the week, resulting in every day of the week having a [Weekend].[True] or [Weekend}.[False} attribute.
  - Assigning attributes for special days or holidays that you designate.
  - Assigning text attributes such as “Monday” and “Tuesday” to appropriate weekdays.

The Administration Services Create Date-Time Dimension wizard builds all members for the calendars and time ranges specified.

Preparing for Creating Date-Time Dimensionality

Before running the Create Date-Time Dimension wizard, consider the following points:

- What time period must the database model? You must provide the start and end dates.
- What type of calendar fits your business use?

  See “Understanding Date-Time Calendars” on page 946.

- What time-period granularity should the database reflect in its outline? Does business analysis drill down to the week level, or can weeks be omitted because reporting is by month or a higher level? Are any business measures analyzed and reported down to the day level? Are semester or trimester breakdowns required?
- Would analysis of data within time relationships be valuable? For example, within a business calendar containing quarters, would the ability to see trends for each closing quarter be useful, such as whether sales increase toward the close of each?

The Create Date-Time Dimension wizard is a powerful tool with many options. Consider some practice runs to better understand the dimensions it builds. See also the topics for each page of this wizard in *Oracle Essbase Administration Services Online Help*.

Understanding Date-Time Calendars

The Create Date-Time Dimension wizard calendar templates determine the hierarchical structure of members in the date-time type of dimension. For all calendars, you must select a date-time dimension name, the calendar type, the first day of the week, and the beginning and ending dates for which you want to build date-time dimension members. You also must select time depths, which determine the calendar member hierarchy. The year time depth is required for all calendars. The availability of other time depths depends on the calendar type and other time depths selected. For example, semester and trimester time depths are mutually exclusive.

If a time depth is selected, the wizard creates, within each year, a member for each instance of that time depth. For example, if the month time depth is selected, 12 month members are created in the hierarchy under each year. A naming rule indicates a naming pattern for each member, which includes a number for the relative position within its year ancestor. For example, in the Gregorian calendar shown in Figure 170 on page 944, the month members follow a default naming pattern that includes the month number such as "Month 4 of Gregorian Year 2008."

Note:

When selecting naming rules for time dimensions, it is possible to create duplicate members. This can cause the create date-time dimension procedure to fail unless the outline is set to allow duplicate member names. If the outline is not set to allow duplicate member names, you must avoid duplicates.

Depending on the calendar template and chosen time depths, you may need to define year, month, or period characteristics (semantic rules) such as period starting or ending dates, week count, or how months or weeks are grouped within their parent members. These characteristics affect the rules by which members are built and named. For some calendars, availability of year, period, or month semantic rules is dependent on other year, period, or month options selected. See "Create Date-Time Dimension Wizard—Select Calendar Hierarchy Panel" in *Oracle Essbase Administration Services Online Help*.

The Create Date-Time Dimension wizard provides templates for the following calendar types:

Gregorian

The Gregorian calendar is the standard 12-month calendar, January 1–December 31. Time depths are year, semester, trimester, quarter, month, week, and day.

Fiscal

Fiscal calendar definitions are based on company financial reporting requirements and can start on any date. Weeks In fiscal calendars have 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, (4-4-5, 4-5-4, or 5-4-4 weeks). If the year has 53 weeks, one month can have an extra week.

The week definition determines how to divide the calendar year into weeks. You can adjust the week definition to make a 52 week or 53 week year. Time depths are year, semester, trimester, quarter, month, week, and day.

Retail

The retail calendar, from the National Retail Federation, is modeled to analyze week-over-week data across years. It has a 4-5-4 quarter pattern with leap weeks every five to six years. The starting date differs from year to year but always falls in early February. When comparing year over year, standard practice is to omit the first week of a 53-week year to normalize for the extra week while the years keep the same holidays. Available time depths are year, semester, quarter, month, week, and day.

Manufacturing

The manufacturing calendar defines a 13-period year, made up of seven-day weeks. Manufacturing periods are divided into three quarters of three periods each and one quarter of four periods. All periods have four weeks, except for 53-week years, in which one period has five 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 weeks or 53. If the year has 52 weeks, you may need to specify both 52-week and 53-week options, to pare the weeks first down to 53, then pare them down to 52. Available time depths are year, semester, quarter, period, week, and day.

ISO 8601

The ISO 8601 calendar contains seven-day weeks. The year can start before or after January 1 and is modeled to start on a day such that the first week of the ISO calendar contains the first Thursday of January. The first day of the week is Monday. Year, week, and day time depths are required for the ISO calendar, with no additional time-depth options.

Modifying or Deleting Date-Time Dimensions

The only way to modify a date-time dimension is manually. Because of its tight structure, modifying a date-time dimension can be risky. In most cases, it is better to delete the dimension and use the Create Date-Time Dimension wizard to recreate it with the changes built in by the wizard, particularly if changes involve adding or removing members.

Even for minor changes, you must be aware of the comments structure described in "Understanding Date-Time Dimensions" on page 942. The TI-HIER and TI-MBR,TSPAN sections must be at the beginning of the comments field. You can add comments after the last ! mark, such as:

```
TI-MBR,TSPAN[1-1-2006,1-31-2006]! This is the First Month
```

Note:

If you edit the TI-HIER and TI-MBR,TSPAN sections in member comments, you must ensure that the correct members exist throughout the hierarchy, and you must edit dependent ranges for ancestors or descendants. For example, if you change the span of a given month, you must also change the span of the quarter.

If you delete a date-time dimension, also delete all associated attribute dimensions. Linked attributes have meaning only when they are whole; that is, all members exist and be associated in the correct sequence. An outline is invalid if you disassociate a linked attribute from only a few members.

Be aware of the verification rules summarized in the following sections.

Verification Rules for Date-Time Dimensions

- The dimension is tagged as "Date-Time".
- The dimension is a single-hierarchy, stored dimension.
- The hierarchy is balanced; all level 0 members are of the same generation.
- Members are all tagged with the addition (+) consolidator.
- Member comment fields for each member contain the following information in the left-most part of the comment. (Details are described in "Understanding Date-Time Dimensions" on page 942.)
    - The member comment field for the top dimension member includes contains the TI-HIER, TP[ ] specification that contains an array that describes the hierarchy structure.
    - The member comment field for each member including the top dimension member contains the TI-MBR, TSPAN[ ] specification that includes the date range for the member, with starting date preceding ending date.
- Along a generation, date ranges should be contiguous and should increase in ascending order. For example, along the month generation, the date range for each month member must have a starting date that logically follows from the ending date of the previous period.

There must be no time gaps, and the span of the parent member must equal the total time span of its children.

Verification Rules for Linked Attribute Dimensions

- Linked attribute dimensions are tagged as "Linked Attribute."
- Linked attribute dimensions are associated with the date-time dimension.
- Each linked attribute dimension has two levels only, the parent level and the children attribute members.
- Linked attribute member names are sequential numbers, starting with 1.
- If attribute dimensions exist, all combinations of attachment and association levels for the date-time dimension must exist. For example, if the date-time dimension includes year, semester, and month time depths, if you create linked attribute dimensions you must create "Month by Year," "Month by Semester," and "Semester by Year."

Loading Data Mapped to Dates

You can load data into the level 0 members of date-time dimensions using date strings instead of member names. Even if the date hierarchy does not span to the day granularity level, the data source can be specified by individual dates. The load process aggregates the values and stores them at the appropriate level.

Loading data based on date provides the following advantages:

- Data can be loaded from any day-level load file, as long as the data file date format is supported.
- If you set the data load to "Add to existing cells," data can be loaded from a day-level load file to a less granular level in the hierarchy; for example to week or month level 0 cells.

Table 132 lists the date format strings you can use when you define the date field in the data load rules file:

Table 132 Date Format Strings

Date Format String	Example
mon dd yyyy	Jan 15 2006
Mon dd yyyy	January 15 2006
mm/dd/yy	01/15/06
mm/dd/yyyy	01/15/2006
yy.mm.dd	06.01.06
dd/mm/yy	15/01/06
dd.mm.yy	15.01.06

Date Format String	Example
dd-mm-yy	15-01-06
dd Mon yy	15 January 06
dd mon yy	15 Jan 06
Mon dd yy	January 15 06
mon dd yy	Jan 15 06
mm-dd-yy	01-15-06
yy/mm/dd	06/01/15
yymmdd	060115
dd Mon yyyy	15 January 2006
dd mon yyyy	15 Jan 2006
yyyy-mm-dd	2006-01-15
yyyy/mm/dd	2006/01/15
Long Name	Sunday, January 15, 2006
Short Name	1/8/06 (m/d/yy)

Note:

Using extra white space not included in the internal format strings returns an error.

Note:

Trailing characters after the date format has been satisfied are ignored. If you erroneously use a date string of 06/20/2006 with date format mm/dd/yy, the trailing 06 is ignored and the date is interpreted as June 20, 2020.

Note:

Long Name format is not verified for a day-of-week match to the given date.

Analyzing Time-Based Data

The MDX and Smart View interfaces can take advantage of linked attributes for analyzing periodic relationships within data.

Using Smart View to Analyze Time-Related Data

Oracle Hyperion Smart View for Office, Fusion Edition provides use of linked attributes in several different ways:

- Working on the free-form grid. With a date-time member on the grid selected, using Member Selection shows Date-Time as a dimension type and lists the members of that dimension. The Period and Range filters enable you to select and display information using linked attributes.
- Using Query Designer.
  - Selecting Date-Time dimension on the Query Designer Toolbar includes linked attributes in the attributes list.
  - Using the POV toolbar, you can select a linked attribute, drag it to the free-form grid, and execute the query.

For additional information, see *Oracle Hyperion Smart View for Office Online Help*.

Analyzing Time-Based Metrics with MDX

Table 133 lists the MDX functions that are provided for analysis of date hierarchies. For a complete description of these functions, see the *Oracle Essbase Technical Reference*.

Table 133 MDX Functions for Analyzing Date Hierarchies

Function	Description
DateDiff	Returns the difference between two input dates.
DatePart	Returns the date part (Year/Quarter/Month/Day/DayOfYear/Weekday) as a number.
DateRoll	To the given date, rolls (adds or subtracts) a number of specific time intervals, returning another date.
DateToMember	Returns the date-hierarchy member specified by the input date.
FormatDate	Returns a formatted date-string.
GetFirstDate	Returns the start date for a date-hierarchy member.
GetLastDate	Returns the end date for a date-hierarchy member.
Today	Returns a number representing the current date on the Essbase computer.
ToDateEx	Converts any date string to a date that can be used in calculations.

60 Loading, Calculating, and Retrieving Aggregate Storage Data

In This Chapter

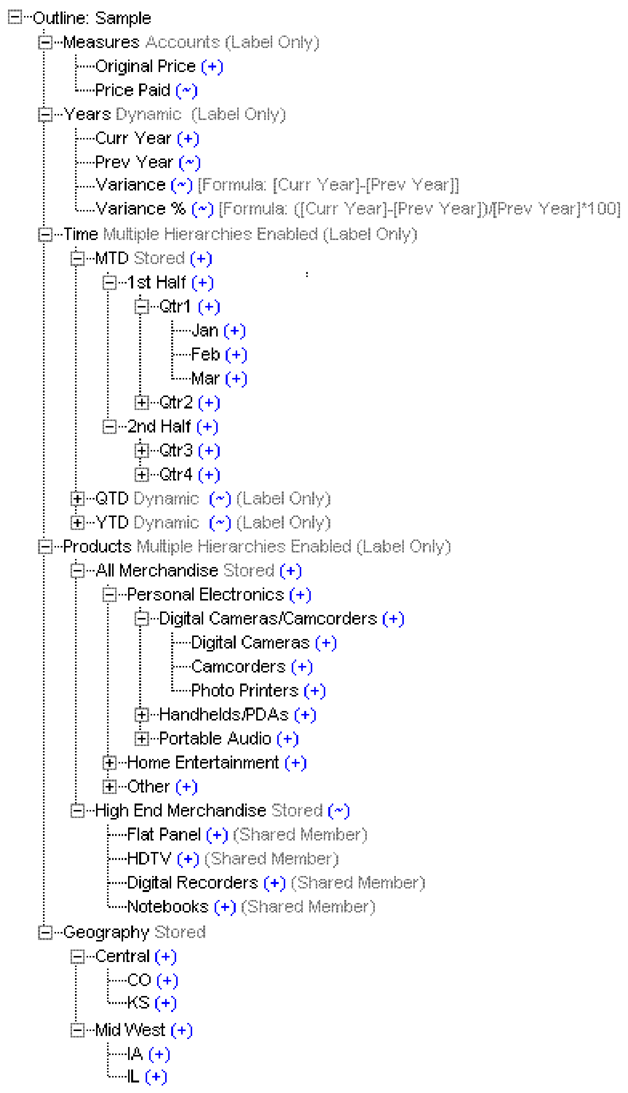
The information in this chapter applies only to aggregate storage databases and is not relevant to block storage databases. Also see Chapter 57, "Comparison of Aggregate and Block Storage."

Introduction

The most common processes for working with database information include maintaining the outline, loading data values to the database, calculating values, and retrieving database information. Performing these tasks with aggregate storage databases is different from performing them with block storage databases.

Examples in this chapter refer to the outline in Figure 171.

Figure 171 Sample Aggregate Storage Outline



```
Outline: Sample
  Measures Accounts (Label Only)
    Original Price (+)
    Price Paid (~)
  Years Dynamic  (Label Only)
    Curr Year (+)
    Prev Year (~)
    Variance (~) [Formula: [Curr Year]-[Prev Year]]
    Variance % (~) [Formula: ([Curr Year]-[Prev Year])/[Prev Year]*100]
  Time Multiple Hierarchies Enabled (Label Only)
    MTD Stored (+)
      1st Half (+)
        Qtr1 (+)
          Jan (+)
          Feb (+)
          Mar (+)
        Qtr2 (+)
      2nd Half (+)
        Qtr3 (+)
        Qtr4 (+)
    QTD Dynamic  (~) (Label Only)
    YTD Dynamic  (~) (Label Only)
  Products Multiple Hierarchies Enabled (Label Only)
    All Merchandise Stored (+)
      Personal Electronics (+)
        Digital Cameras/Camcorders (+)
          Digital Cameras (+)
          Camcorders (+)
          Photo Printers (+)
        Handhelds/PDAs (+)
        Portable Audio (+)
      Home Entertainment (+)
      Other (+)
    High End Merchandise Stored (~)
      Flat Panel (+) (Shared Member)
      HDTV (+) (Shared Member)
      Digital Recorders (+) (Shared Member)
      Notebooks (+) (Shared Member)
  Geography Stored
    Central (+)
      CO (+)
      KS (+)
    Mid West (+)
      IA (+)
      IL (+)
```

The simplified aggregate storage outline in Figure 171 is not completely expanded. A plus sign (+) node at the left of a member name indicates that the member has children that are not displayed.

Preparing Aggregate Storage Databases

The topics in this section describe dimension build and data load process differences between aggregate storage and block storage databases. You should be familiar with data load, dimension build, and rules file concepts and procedures.

For information about using data sources to change outlines and to load data values, see Chapter 16, "Understanding Data Loading and Dimension Building" and Chapter 19, "Performing and Debugging Data Loads or Dimension Builds."

For information on the maximum size of a buildable aggregate storage outline, see "Outline Paging Limits" on page 927.

Building Dimensions in Aggregate Storage Databases

Aggregate storage dimension build changes to the outline can result in all aggregate views or all data values being cleared from the database when the dimension build is finished. "Aggregate Storage Database Restructuring" on page 994 describes the results of outline changes.

If you use more than one data source to build dimensions, you can save processing time by performing an incremental dimension build. Incremental dimension builds enable you to defer restructuring until all data sources have been processed. For information about incremental dimension build, see "Performing Deferred-Restructure Dimension Builds" on page 299.

Differences between outline characteristics of block storage outlines and aggregate storage outlines affect data sources and rules files. For example, defining a dimension as sparse or dense is not relevant to aggregate storage outlines.

Rules File Differences for Aggregate Storage Dimension Builds

Rules files for building aggregate storage outlines must define only outline properties and field types that apply to aggregate storage outlines. See Table 123, "Outline Differences Between Aggregate Storage and Block Storage," on page 909.

After converting a block storage outline to aggregate storage, update the rules files by associating them to the aggregate storage version of the outline. See "Associating an Outline with an Editor" in the *Oracle Essbase Administration Services Online Help*.

Field Type Differences for Aggregate Storage Dimension Builds

The following table shows the field types that are available only on aggregate storage databases:

Table 134 Field Types for Aggregate Storage Dimension Builds

Field Type	What the Field Contains	Valid Build Methods
Solve order	A number (0–127) that specifies the order in which the member is evaluated in the outline. Values less than 0 or greater than 127 are reset to 0 and 127, respectively. No warning message is displayed. For information on the solve order property, see "Calculation Order" on page 973.	Generation, level, and parent-child references

➤ To set field information, see "Setting Field Types" in the *Oracle Essbase Administration Services Online Help*.

Data Prep Editor Dialog Boxes for Aggregate Storage

As you edit rules files for aggregate storage databases, some dimension build rules file options that apply only to block storage databases are displayed in Data Prep Editor dialog boxes. Table 135 lists block storage rules file settings that do not apply to aggregate storage outlines. Entries in rules files for these options are ignored when the rules file is processed.

Table 135 Aggregate Storage Dimension Build Rules File Differences

Rules File Location in the Administration Services Interface	Dimension Build Rules File Options That Do Not Apply to Aggregate Storage Databases
Dimension Build Settings dialog box, Global Settings tab	Data configuration options
Dimension Properties dialog box, Dimension Properties tab	Dimension types option: Country Two-Pass calculation option Data storage options: ● Dynamic Calc and Store ● Dynamic Calc All configuration options
Dimension Properties dialog box, Accounts Dimension tab	None of the options on this tab apply.
Field Properties dialog box, Dimension Build Properties tab	Field type options: ● Currency name ● Currency category Currency functionality does not apply to aggregate storage databases.

See the *Oracle Essbase Administration Services Online Help*.

Data Source Differences for Aggregate Storage Dimension Builds

Data sources for modifying aggregate storage outlines should not include field values that apply only to block storage outlines. Table 136 lists the member codes that are recognized in dimension build data sources as properties for members of aggregate storage database outlines. Any other consolidation code is ignored and + (Add) is assumed. See "Using the Data Source to Work with Member Properties" on page 275.

Table 136 Valid Consolidation Properties for Members of Aggregate Storage Outlines

Code	Description
%	Express as a percentage of the current total in a consolidation (applies only to members of a dynamic hierarchy)
*	Multiply by the current total in a consolidation (applies only to members of a dynamic hierarchy)
+	Add to the current total in a consolidation (applies only to members of a dynamic hierarchy)
-	Subtract from the current total in a consolidation (applies only to members of a dynamic hierarchy)
/	Divide by the current total in a consolidation (applies only to members of a dynamic hierarchy)
~	Exclude from the consolidation (applies only to members of a dynamic hierarchy or members beneath Label Only members in a stored hierarchy)
O	Tag as label only
N	Never allow data sharing

Code	Description
C	Set member as top of a stored hierarchy (applies to dimension member or generation 2 member)
D	Set member as top of a dynamic hierarchy (applies to dimension member or generation 2 member)
H	Set dimension member as multiple hierarchies enabled (applies to dimension member only)

Currency name and currency category field types are not supported for aggregate storage outlines.

In aggregate storage outlines, formulas must be specified in the same format as MDX numeric value expressions. See "Developing Formulas on Aggregate Storage Outlines" on page 930.

Building Alternate Hierarchies in Aggregate Storage Databases

To build shared members in an aggregate storage outline, select the autoconfigure setting in the Dimension Build Settings dialog box in Administration Services, and then make sure that Essbase is set up to allow sharing (clear "Do Not Share" in the Dimension Build Settings dialog box in Administration Services or omit the "Never allow Data Sharing" property in the rules file). When autoconfigure is selected and sharing is enabled, Essbase automatically creates duplicate members under a new parent as shared members. See "Building Shared Members by Using a Rules File" on page 330.

Note:

There are restrictions on using the duplicate generation (DUPGEN) method to build alternate hierarchies in aggregate storage outlines. See Table 126, "Data Load Differences Between Aggregate Storage and Block Storage," on page 911.

Caution!

In ASO alternate hierarchies, you can associate attributes only with upper level (level 0) members. If attributes are associated with non-level 0 members, Essbase returns this error message: `Member [XXX] in alternate hierarchy has attribute association.`

Loading Data into Aggregate Storage Databases

For general information about loading data, see Chapter 19, "Performing and Debugging Data Loads or Dimension Builds."

Aggregate storage databases facilitate analysis of very large dimensions containing up to a million or more members. To efficiently support loading data values into such large databases, Essbase:

- Allows the processing of multiple data sources through temporary aggregate storage data load buffers
- Allows you to control the percentage of resources a data load buffer uses

- Allows an aggregate storage database to contain multiple slices of data (a query to the database accesses each slice, collecting all of the data cells)
- Provides an incremental data load process that completes in a length of time that is proportional to the size of the incremental data

➤ To load values to an aggregate storage databases, use a tool:

Tool	Topic	Location
Administration Services	Performing a Data Load or Dimension Build for Aggregate Storage Databases	*Oracle Essbase Administration Services Online Help*
MaxL	**alter database** **import data**	*Oracle Essbase Technical Reference*
ESSCMD	IMPORT LOADDB	*Oracle Essbase Technical Reference*

Note:

If values have been calculated and stored through an aggregation, Essbase automatically updates higher-level stored values when data values are changed. No additional calculation step is necessary. The existence and size of an aggregation can affect the time it takes to perform a data load. See "Aggregations" on page 978.

Note:

You cannot export data when loading data into a database.

Incrementally Loading Data Using a Data Load Buffer

Using the **import database data** MaxL statement to load data values from a single data source does not involve the aggregate storage data load buffer.

If you use multiple **import database data** MaxL statements to load data values to aggregate storage databases, you can significantly improve performance by loading values to a temporary data load buffer first, with a final write to storage after all data sources have been read.

In the aggregate storage data load buffer, Essbase sorts and commits the values after all data sources have been read. If multiple (or duplicate) records are encountered for any specific data cell, the values are accumulated (see "Resolving Cell Conflicts" on page 962). Essbase then stores the accumulated values—replacing, adding to, or subtracting from existing data values in the database. Using the aggregate storage data load buffer can significantly improve overall data load performance.

Note:

When using the aggregate storage data load buffer, the choice for replacing, adding, or subtracting values is specified for the entire set of data sources when loading the data buffer contents to the database.

While the data load buffer exists in memory, you cannot build aggregations or merge slices, because these operations are resource-intensive. You can, however, load data to other data load buffers, and perform queries and other operations on the database. There might be a brief wait for queries, until the full data set is committed to the database and aggregations are created.

The data load buffer exists in memory until the buffer contents are committed to the database or the application is restarted, at which time the buffer is destroyed. Even if the commit operation fails, the buffer is destroyed and the data is not loaded into the database. (You can manually destroy a data load buffer by using the **alter database** MaxL statement. See the *Oracle Essbase Technical Reference*.)

Note:

Stopping the application before committing the buffer contents destroys the buffer. In this situation, after restarting the application, you must initialize a new buffer and load the data to it.

➤ To use the data load buffer for aggregate storage databases:

1. Prepare the data load buffer, where data values are sorted and accumulated by using the **alter database** MaxL statement to initialize an aggregate storage data load buffer. For example:

   ```
   alter database AsoSamp.Sample
      initialize load_buffer with buffer_id 1;
   ```

2. Load data from your data sources into the data load buffer using the **import database** MaxL statement. Use multiple statements to load data from multiple data sources. You can include any combination of data sources, such as `.xls` files, text files, and SQL relational sources. Specify a rules file if the data source requires one.

   The following example loads three data sources, one of which uses a rules file, into the same data load buffer:

   ```
   import database AsoSamp.Sample data
      from server data_file 'file_1.txt'
      to load_buffer with buffer_id 1
      on error abort;
   import database AsoSamp.Sample data
      from server data_file 'file_2'
      using server rules_file ‘rule’
      to load_buffer with buffer_id 1;
      on error abort;
   import database AsoSamp.Sample data
      from server excel data_file 'file_3.xls'
      to load_buffer with buffer_id 1
      on error abort;
   ```

To load data into multiple load buffers at the same time, see "Performing Multiple Data Loads in Parallel" on page 962.

3. Use the **import database** MaxL statement to commit the data load buffer contents to the database. For example:

   ```
   import database AsoSamp.Sample data
      from load_buffer with buffer_id 1;
   ```

   To commit the contents of multiple data load buffers into the database with one MaxL statement, see "Performing Multiple Data Loads in Parallel" on page 962.

The following incremental data load example provides optimal performance when new data values do not intersect with existing values:

1. Create a single data load buffer using the ignore_missing_values and ignore_zero_values properties. For example:

   ```
   alter database AsoSamp.Sample
      initialize load_buffer with buffer_id 1
      property ignore_missing_values, ignore_zero_values;
   ```

   If the database must be available for send data requests while the database is being updated, initialize the data load buffer with the **resource_usage** grammar set for 80%. For example:

   ```
   alter database AsoSamp.Sample
      initialize load_buffer with buffer_id 1
      resource_usage 0.8 property
      ignore_missing_values, ignore_zero_values;
   ```

2. Load the data into the buffer. For example:

   ```
   import database AsoSamp.Sample data
      from server data_file 'file_1.txt'
      to load_buffer with buffer_id 1
      on error abort;
   import database AsoSamp.Sample data
      from server data_file 'file_2'
      to load_buffer with buffer_id 1;
      on error abort;
   ```

3. Commit the contents of the data load buffer to the database by creating a slice and adding values. For example:

   ```
   import database AsoSamp.Sample data
      from load_buffer with buffer_id 1
      add values create slice;
   ```

Controlling Data Load Buffer Resource Usage

When performing an incremental data load, Essbase uses the aggregate storage cache for sorting data. You can control how much of the cache a data load buffer can use by specifying the percentage (between .01 and 1.0 inclusive; only two digits after the decimal point are significant —for example, 0.029 is interpreted as 0.02). By default, the resource usage of a data load buffer is set to 1.0, and the total resource usage of all data load buffers created on a database cannot exceed 1.0. For example, if a buffer of size 0.9 exists, you cannot create another buffer of a size greater than 0.1.

Note:

Send operations internally create load buffers of size 0.2; therefore, a load buffer of the default size of 1.0 will cause send operations to fail because of insufficient data load buffer resources.

To set the share of resources the buffer is allowed to use, use the **alter database** MaxL statement with the **resource_usage** grammar.

For example, to set the resource_usage to 50% of the total cache, use this statement:

```
alter database AsoSamp.Sample
   initialize load_buffer with buffer_id 1
   resource_usage .5;
```

Setting Data Load Buffer Properties

When loading data incrementally, you can specify how missing and zero values in the source data are treated when loading the data into the data load buffer.

For resolving cell conflicts for duplicate cells, you can specify whether to use the last cell loaded into the load buffer.

The data load buffer properties:

- ignore_missing_values: Ignores #MI values in the incoming data stream
- ignore_zero_values: Ignores zeros in the incoming data stream
- aggregate_use_last: Combines duplicate cells by using the value of the cell that was loaded last into the load buffer

  Note:

  When loading text and date values into an aggregate storage database, use the aggregate_use_last option to help eliminate invalid aggregations. For other guidelines, see "Loading, Clearing, and Exporting Text and Date Measures" on page 192.

If you use multiple properties in the command and any conflict, the last property listed takes precedence.

To set data load buffer properties, use the **alter database** MaxL statement with the **property** grammar.

For example:

```
alter database AsoSamp.Sample
   initialize load_buffer with buffer_id 1
   property ignore_missing_values, ignore_zero_values;
```

In Administration Services Console, you can set options to ignore missing values and zero values, and to aggregate duplicate cells by using the value of the last cell loaded, on the Data Load dialog box. See "Data Load Dialog Box" in the *Oracle Essbase Administration Services Online Help*.

Resolving Cell Conflicts

By default, when cells with identical keys are loaded into the same data load buffer, Essbase resolves the cell conflict by adding the values together.

To create a data load buffer that combines duplicate cells by accepting the value of the cell that was loaded last into the load buffer, use the **alter database** MaxL statement with the **aggregate_use_last** grammar.

For example:

```
alter database AsoSamp.Sample
   initialize load_buffer with buffer_id 1
   property aggregate_use_last;
```

Note:

When using data load buffers with the **aggregate_use_last** grammar, data loads are significantly slower, even if there are not any duplicate keys.

Performing Multiple Data Loads in Parallel

Multiple data load buffers can exist on an aggregate storage database. To save time, you can load data into multiple data load buffers at the same time.

Although only one data load commit operation on a database can be active at any time, you can commit multiple data load buffers in the same commit operation, which is faster than committing buffers individually.

Note:

When using Administration Services Console to load data into an aggregate storage database, only a single data load buffer is used.

To load data into multiple data load buffers at the same time, use separate MaxL Shell sessions. For example, in one MaxL Shell session, load data into a buffer with an ID of 1:

```
alter database AsoSamp.Sample
   initialize load_buffer with buffer_id 1 resource_usage 0.5;
import database AsoSamp.Sample data
   from data_file "dataload1.txt"
   to load_buffer with buffer_id 1
   on error abort;
```

Simultaneously, in another MaxL Shell session, load data into a buffer with an ID of 2:

```
alter database AsoSamp.Sample
   initialize load_buffer with buffer_id 2 resource_usage 0.5;
import database AsoSamp.Sample data
   from data_file "dataload2.txt"
   to load_buffer with buffer_id 2
   on error abort;
```

When the data is fully loaded into the data load buffers, use one MaxL statement to commit the contents of both buffers into the database by using a comma-separated list of buffer IDs:

For example, this statement loads the contents of buffers 1 and 2:

```
import database AsoSamp.Sample data
   from load_buffer with buffer_id 1, 2;
```

Note:

When loading SQL data into aggregate storage databases, you can use up to eight rules files to load data in parallel. This functionality is different than the process described above. When preforming multiple SQL data loads in parallel, you use one **import database** MaxL statement with the **using multiple rules_file** grammar. Essbase initializes multiple temporary aggregate storage data load buffers (one for each rules file) and commits the contents of all buffers into the database in one operation. See the *Oracle Essbase SQL Interface Guide*.

Listing Data Load Buffers for an Aggregate Storage Database

Multiple data load buffers can exist on an aggregate storage database. For a list and description of the data load buffers that exist on an aggregate storage database, use the **query database** MaxL statement with the **list load_buffers** grammar:

```
query database appname.dbname list load_buffers;
```

This statement returns the following information about each existing data load buffer:

Field	Description
buffer_id	ID of a data load buffer (a number between 1 and 4,294,967,296).
internal	A Boolean that specifies whether the data load buffer was created internally by Essbase (TRUE) or by a user (FALSE).
active	A Boolean that specifies whether the data load buffer is currently in use by a data load operation.
resource_usage	The percentage (a number between .01 and 1.0 inclusive) of the aggregate storage cache that the data load buffer is allowed to use.
aggregation method	One of the methods used to combine multiple values for the same cell within the buffer: ● AGGREGATE_SUM: Add values when the buffer contains multiple values for the same cell. ● AGGREGATE_USE_LAST: Combine duplicate cells by using the value of the cell that was loaded last into the load buffer.
ignore_missings	A Boolean that specifies whether to ignore #MI values in the incoming data stream.
ignore_zeros	A Boolean that specifies whether to ignore zeros in the incoming data stream.

Creating a Data Slice

You can incrementally commit the data load buffer to an aggregate storage database to create a slice. After loading the new slice into the database, Essbase creates all necessary views on the slice (such as aggregate views) before the new data is visible to queries.

To create a data slice, use the **import database** MaxL statement with the **create slice** grammar.

For example, to create a slice by overriding values (the default), use this statement:

```
import database AsoSamp.Sample data
   from load_buffer with buffer_id 1
   override values create slice;
```

Note:

If you use override values when creating a slice, #MISSING values are replaced with zeros. Using this option is significantly slower than using the add values or subtract values options.

In Administration Services Console, you can set an option to create a slice on the Data Load dialog box. See "Data Load Dialog Box" in the *Oracle Essbase Administration Services Online Help*.

Merging Incremental Data Slices

You can merge all incremental data slices into the main database slice or merge all incremental data slices into a single data slice while leaving the main database slice unchanged. To merge slices, you must have the same privileges as for loading data (Administrator or Database Manager permissions).

When the data for a new data slice is in the data load buffer, Essbase scans the list of incremental data slices and considers whether to automatically merge them with the new slice. To qualify for an automatic merge, a slice must be smaller than 100,000 cells or smaller than two times the size of the new slice. Slices larger than 5,000,000 cells are never automatically merged. For example, if a new slice contains 300,000 cells, incremental data slices that contain 90,000 cells and 500,000 cells are automatically merged with the new cell. An incremental data slice that contains 700,000 is not.

After the new input view is written to the database, Essbase creates the aggregate views for the slice. The views created for the new slice are a subset of the views that exist on the main database slice.

Note:

You cannot export data when performing a merge.

If you cleared data from a region using the logical clear region operation, which results in a value of zero for the cells you cleared, you can elect to remove zero value cells during the merge operation. See "Clearing Data from Specific Regions of Aggregate Storage Databases" on page 968.

To perform merging operations, use the **alter database** MaxL statement with the **merge** grammar.

For example, to merge all incremental data slices into the main database slice, use this statement:

```
alter database AsoSamp.Sample
   merge all data;
```

To merge all incremental data slices into the main database slice and remove zero value cells, use this statement:

```
alter database AsoSamp.Sample
   merge all data remove_zero_values;
```

To merge all incremental data slices into a single data slice, use this statement:

```
alter database AsoSamp.Sample
   merge incremental data;
```

In Administration Services Console, you can merge data slices. See "Merging Incremental Data Slices (Aggregate Storage Databases)" in the *Oracle Essbase Administration Services Online Help*.

Replacing Database or Incremental Data Slice Contents

For data sets that are small enough to reload completely while maintaining low data latency, Essbase can remove the current contents of a database and replace the database with the contents of a specified data load buffer. The atomic replacement functionality transitions querying the old contents of the database to the new contents without interrupting service. The newly loaded data set is aggregated to create the same set of views that existed for the replaced data set.

Essbase also allows for automatically replacing the contents of all incremental data slices in a database. Consider a situation in which data can be separated into a relatively large, static data set that is never updated and a relatively small, volatile data set for which the individual updates are difficult to identify but are confined to the volatile data set. For example, the large, static data set consists of historical transaction data for the last three years; however, for the transaction data for the past two months, users can change a characteristic of a transaction in the source database. Tracking these changes can be prohibitively complex. You can load the static data set as the main slice in a database and the volatile data set as one or more incremental slices.

Essbase removes the current contents of all incremental data slices and creates a new slice (using the **add values** grammar) with the contents of a specified data load buffer. The newly loaded data set is augmented with aggregated views based on the set of views that exist on the main slice.

Note:

To use the **override** grammar, create a data load buffer with the ignore_missing_values property for optimal performance. Additionally, you must ensure that there are not any conflicts between the static and volatile data sets (for example, there should not be a value in each data set for the same cell).

To replace the contents of a database or the incremental data slices in a database, use the **import database** MaxL statement with the **override** grammar.

For example, to replace the contents of a database, use this statement:

```
import database AsoSamp.Sample data
   from load_buffer with buffer_id 1
   override all data;
```

To replace the contents of all incremental data slices with a new slice, use this statement:

```
import database AsoSamp.Sample data
   from load_buffer with buffer_id 1
   override incremental data;
```

Note:

If the override replacement fails, Essbase continues to serve the old data set.

In Administration Services Console, you can set the option to replace the contents of the database or data slices in the Data Load dialog box. See "Replacing Database or Data Slice Contents (Aggregate Storage Database)" in the *Oracle Essbase Administration Services Online Help*.

Viewing Incremental Data Slices Statistics

Essbase provides statistics on the size and number of incremental data slices, and the cost of querying the incremental data slices.

The time it takes for a query to access all of the incremental data slices is expressed as a percentage (between .01 and 1.0 inclusive). If a database has a main slice and multiple incremental data slices, a query statistic of 0.66 means that two-thirds of the query time was spent querying the incremental data slices and one-third was spent querying the main data slice. If the cost of querying the incremental data slices is too high, you can merge the slices.

To view the incremental data slice statistics, see "Viewing Aggregate Storage Statistics" in the *Oracle Essbase Administration Services Online Help*.

Managing Disk Space For Incremental Data Loads

Incremental data loads on aggregate storage databases may use disk space up to two times the size of your current data files. In cases where databases are larger than 2 GB, you can reduce disk space utilization by setting the maximum file size of the default tablespace to no more than 2 GB. See "Working with Tablespaces" on page 992.

➤ To set the maximum file size of the default tablespace, use a tool:

Tool	Topic	Location
Administration Services	Managing Tablespaces	*Oracle Essbase Administration Services Online Help*
MaxL	**alter tablespace**	*Oracle Essbase Technical Reference*

Using Spreadsheet Add-in

In Spreadsheet Add-in, the send command is equivalent to using the incremental data load functionality with the **override** grammar.

While performing a send operation, new requests for lock, unlock, and retrieve and lock will wait until the send operation is completed.

See the *Oracle Essbase Spreadsheet Add-in Online Help*.

Data Source Differences for Aggregate Storage Data Loads

While processing data source records for loading values into aggregate storage databases, Essbase processes source data records only for the level 0 dimension intersections where the member does not have a formula.

The following example shows a data source with records for only level 0 intersections. The last field contains data values; the other fields are level 0 members of their respective dimensions.

```
Jan, Curr Year, Digital Cameras, CO, Original Price, 10784
Jan, Prev Year, Camcorders, CO, Original Price, 13573
```

Essbase ignores records that specify upper-level members and, at the end of the data load, displays the number of skipped records.

For example, the following record would be skipped because member Mid West is a level 1 member:

```
Jan, Curr Year, Digital Cameras, Mid West, Original Price, 121301
```

Sorting data sources is unnecessary because Essbase Server reads and sorts records internally before committing values to the database.

Rules File Differences for Aggregate Storage Data Loads

Rules file specifications for loading values to aggregate storage databases reflect the aggregate storage data load process.

For block storage data loads, through the rules file, you choose for each data source whether to overwrite existing values, add values in the data source to existing values, or subtract them from existing values.

For aggregate storage data loads using the aggregate storage data load buffer, you make this choice for all data load sources that are gathered into the data load buffer before they are loaded to the database.

To use a rules file that was defined for a block storage outline with an aggregate storage outline, first associate the rules file with the aggregate storage outline. See “Associating an Outline with an Editor” in the *Oracle Essbase Administration Services Online Help*.

Clearing Data from an Aggregate Storage Database

You can selectively clear data or clear all data from an aggregate storage database.

Also see "Clearing Aggregations" on page 986.

Clearing Data from Specific Regions of Aggregate Storage Databases

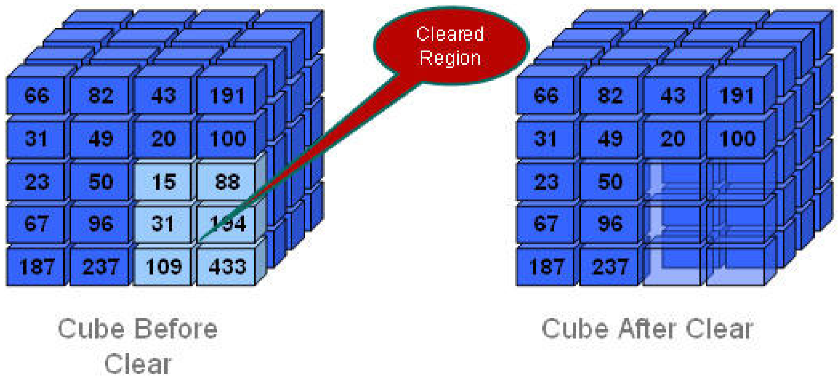
You can clear data from a specified region in an aggregate storage database and retain the data located in other regions. This feature is useful when you want to delete volatile data (such as data corresponding to the last month) but retain historical data. You must have Database Manager or Administrator permission to clear data.

Methods for clearing data from a region:

- Physical

  The input cells in the specified region are physically removed from the aggregate storage database, as illustrated in Figure 172.

  Figure 172 Physically Clearing a Region of Data

  If there are multiple data slices in the database, the physical clear region operation automatically merges all data slices into the main data slice. After data for the specified region is cleared, Essbase materializes all aggregate views that were present in the main data slice before the clear region operation took place.

  The process for physically clearing data completes in a length of time proportional to the size of the input data, not to the size of the data being cleared. Therefore, you might use this method only when removing large slices of data.

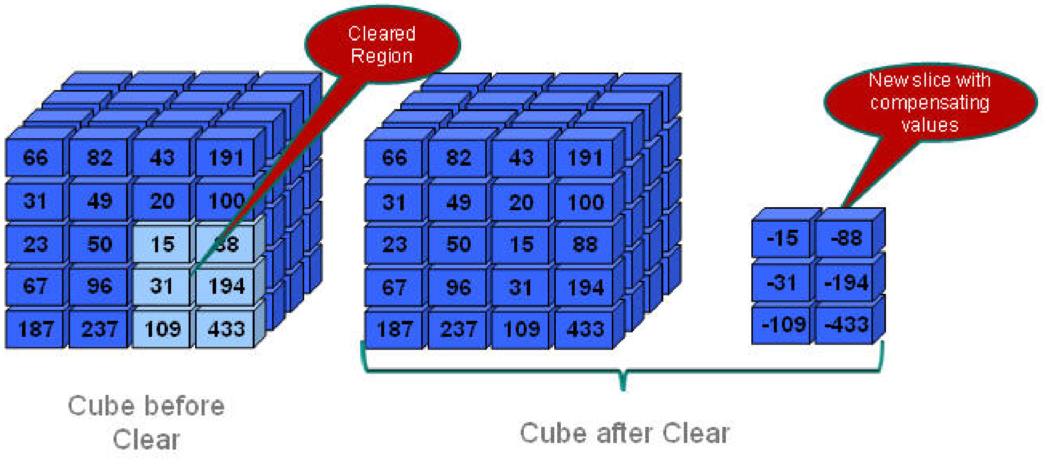
  To physically clear data, use the **alter database** MaxL statement with the **clear data in region** grammar and the **physical** keyword:

  ```
  alter database appname.dbname clear data in region 'MDX set expression'
  physical;
  ```

- Logical

  The input cells in the specified region are written to a new data slice with negative, compensating values that result in a value of zero for the cells you want to clear, as illustrated in Figure 173.

Figure 173 Logically Clearing a Region of Data



The logical clear region operation automatically merges only the data slice with zero values into the main data slice; other data slices in the database are not merged. After data for the specified region is cleared, Essbase materializes aggregate views only in the new data slice.

The process for logically clearing data completes in a length of time that is proportional to the size of the data being cleared. Because compensating cells are created, this option increases the size of the database.

To logically clear data, use the **alter database** MaxL statement with the **clear data in region** grammar but without the **physical** keyword:

```
alter database appname.dbname clear data in region 'MDX set expression';
```

Queries to the logically cleared region return zero values instead of #MISSING values. You may need to update formulas that rely on #MISSING values for empty cells.

To remove cells with a value of zero, use the **alter database** MaxL statement with the **merge** grammar and the **remove_zero_cells** keyword. See the *Oracle Essbase Technical Reference*.

Note:

Oracle does not recommend performing a second logical clear region operation on the same region, because the second operation does not clear the compensating cells created in the first operation and does not create new compensating cells.

In specifying the region to be cleared, follow these guidelines:

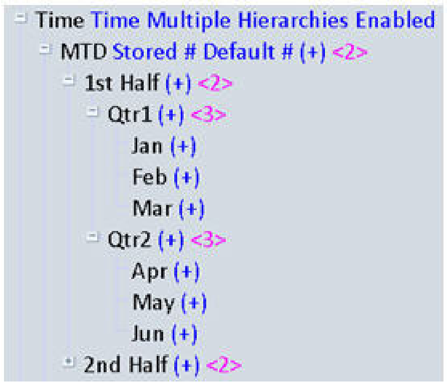
- The region must be symmetrical.
  - {(Jan, Budget)} is a valid symmetrical region that clears all Budget data for Jan.
  - {(Jan, Forecast1),(Feb, Forecast2)} is an invalid region because it consists of two asymmetrical regions (Jan, Forecast1 and Feb, Forecast2).
- Individual members in any dimension in the region specification must be stored members.
- Members in the region cannot be:
  - Dynamic members (members with implicit or explicit MDX formulas)

  - From attribute dimensions

    If you need to clear cells by an attribute, use the Attribute MDX function.

- Members in the region can be upper-level members in stored hierarchies, which is a convenient way to specify multiple level-0 members.

  For example, you can specify Qrt1, which is the same as specifying Jan, Feb, and Mar (the level-0 children of Qrt1):

  ```
  Time Time Multiple Hierarchies Enabled
    MTD Stored # Default # (+) <2>
      1st Half (+) <2>
        Qtr1 (+) <3>
          Jan (+)
          Feb (+)
          Mar (+)
        Qtr2 (+) <3>
          Apr (+)
          May (+)
          Jun (+)
      2nd Half (+) <2>
  ```

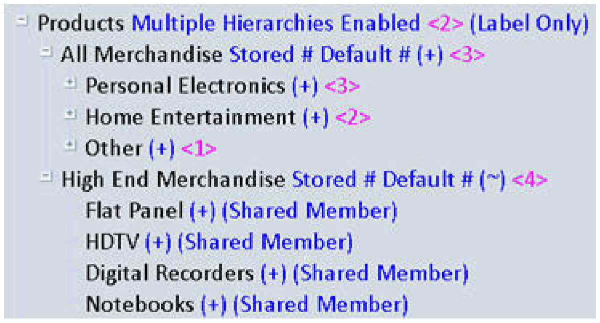
  The following two MaxL statements produce the same results:

  `alter database` *appname*`.`*dbname* `clear data in region '{Qtr1}';`

  `alter database` *appname*`.`*dbname* `clear data in region '{Jan, Feb, Mar}';`

- (Physically clearing data only) Members in the region can be upper-level members in alternate hierarchies.

  For example, you can specify High End Merchandise, which is the same as specifying Flat Panel, HDTV, Digital Recorders, and Notebooks (the shared, level-0 children of High End Merchandise):

  ```
  Products Multiple Hierarchies Enabled <2> (Label Only)
    All Merchandise Stored # Default # (+) <3>
      Personal Electronics (+) <3>
      Home Entertainment (+) <2>
      Other (+) <1>
    High End Merchandise Stored # Default # (~) <4>
      Flat Panel (+) (Shared Member)
      HDTV (+) (Shared Member)
      Digital Recorders (+) (Shared Member)
      Notebooks (+) (Shared Member)
  ```

  The following two MaxL statements produce the same results:

  `alter database` *appname*`.`*dbname* `clear data in region '{High End Merchandise}';`

  `alter database` *appname*`.`*dbname* `clear data in region '{[Flat Panel], [HDTV],[Digital Recorders],[Notebooks]}';`

  To specify members in alternate hierarchies when logically clearing data, use the Descendants MDX function.

Note:

When the region contains upper-level members from alternate hierarchies, you may experience a decrease in performance. In this case, consider using only level-0 members.

- The MDX set expression must be enclosed with single quotation marks.

For example, to clear all January data for Forecast1 and Forecast2 scenarios, use this statement:

```
alter database AsoSamp.Sample clear data in region 'CrossJoin({Jan},
{Forecast1, Forecast2})';
```

During the clear region operation, you cannot perform operations that update the database (such as loading data, merging data slices, or clearing data from another region), nor export data. You can query the database; however, the query results are based on the data set before the clear region operation.

The **clear data in region** grammar cannot clear data from the entire database. See "Clearing All Data from an Aggregate Storage Database" on page 971.

Clearing All Data from an Aggregate Storage Database

Clearing all data from an aggregate storage database is the same as for a block storage database. To clear the entire database, either use:

- The **alter database** MaxL statement with the **reset** grammar:

  ```
  alter database appname.dbname reset;
  ```

- Administration Services Console

  See "Clearing Data" in the *Oracle Essbase Administration Services Online Help*.

Combining Data Loads and Dimension Builds

When using the aggregate storage data load buffer, you can combine data sources and rules files to add members to the outline and to load data values to the level 0 cells. Regardless of the order in which you specify the files, Essbase makes the outline changes and then loads the data values.

Note:

Although dimension builds and data loads for aggregate storage databases can be combined into one operation in Administration Services, Oracle recommends that you separate dimension build operations from data load operations, because some dimension builds clear data, which could lead to data loss. See "Building Dimensions in Aggregate Storage Databases" on page 955.

Calculating Aggregate Storage Databases

Aggregate storage database values are calculated through the outline structure and MDX formulas. When a data load is complete, all the information needed to calculate an aggregate storage database is available. When a retrieval request is made, Essbase Server calculates the needed values by consolidating the values loaded for level 0 members and calculating formulas. Values calculated for retrievals are not stored.

To improve retrieval performance, Essbase can aggregate values and store them ahead of time. However, aggregating and storing all values can be a lengthy process that requires disk space for storage. Essbase provides an intelligent aggregation process that balances time and storage resources. See "Aggregating an Aggregate Storage Database" on page 977.

To prepare an aggregate storage database for retrieval, you create the outline and load the level 0 values. Then you calculate the database by aggregating, and storing additional values, with the remaining values to be calculated when retrieved.

Note:

If a database needs calculation scripts to handle special calculations and data dependencies, consider altering the database design or making it a block storage database.

See Table 124, "Calculation Differences Between Aggregate Storage and Block Storage," on page 910.

Outline Factors Affecting Data Values

The hierarchical structure of an aggregate storage outline determines how values are rolled up. Level 0 member values roll up to level 1 member values, level 1 member values roll up to level 2 member values, and so on.

Consolidation operators assigned to members of dynamic hierarchies define the operations used in the roll-up: add (+), subtract (-), multiply (*), divide (/), percent (%), and no operation (~). For an explanation of operators, see Table 16, "Consolidation Operators," on page 146.

Note:

Members of stored hierarchies can an have only the addition (+) or the no-consolidation (~) operator.

For more complex operations, you can provide MDX formulas on members of dynamic hierarchies. MDX formulas are written in the same format as MDX numeric value expressions. See "Developing Formulas on Aggregate Storage Outlines" on page 930.

Block Storage Calculation Features That Do Not Apply to Aggregate Storage Databases

The following characteristics of calculating block storage databases do not apply to aggregate storage databases:

- Calculation script calculations

- Dynamic Calc and Dynamic Calc and Store member storage properties
- Block storage formula syntax and predefined Essbase functions in formulas
- Custom-defined calculation functions and custom-defined calculation macros
- Formulas on members of dimensions other than members of aggregate storage dynamic hierarchies
- Preloaded values for member intersections above level 0
- Two-pass calculations tags
- Block storage performance features such as Intelligent Calculation

Calculation Order

Aggregate storage calculation order and block storage calculation order differ. For aggregate storage databases, Essbase calculates data in the following order:

1. **Aggregates members of stored hierarchies and attribute dimensions.** The order in which members and dimensions are aggregated is optimized internally and changes according to the nature of the database outline and the existing aggregations. Because the aggregation is additive, the order in which Essbase aggregates the dimensions and members does not affect the results.

   Because the internal aggregation order for an aggregate storage database is not predictable, any inherent rounding errors are also not predictable. These rounding errors are expected behavior in computer calculation and are extremely small in relation to the data values concerned.
2. **Calculates dynamic hierarchy dimension members and formulas.** The order in which members and formulas are evaluated is defined by the solve order property, which you can set for each member or dimension. Calculation order may affect calculation results.

Solve Order Property

The concept of solve order applies to query execution. When a cell is evaluated in a multidimensional query, the order in which the calculations should be resolved may be ambiguous. To remove ambiguity, you can use the solve order property to specify the required calculation priority.

Note:

It is good practice to specify the solve order for each member by setting the solve order property at the member level or at the dimension level. Members without formulas that do not have a specified solve order inherit the solve order of their dimension. Members with formulas that do not have a specified solve order have a solve order of zero.

➤ To specify the solve order for a member or a dimension, use a tool:

Tool	Topic	Location
Administration Services	Specifying the Calculation Order for Members and Dimensions in Aggregate Storage Databases	*Oracle Essbase Administration Services Online Help*
MaxL	**solve_order** parameter in the With section of an MDX query	*Oracle Essbase Technical Reference* (see MDX, Grammar Rules, With Specification)

The value of the solve order property determines the priority with which Essbase calculates the formulas. The formulas on the members that have a specified solve order are calculated in order from the lowest solve order to the highest. (See "Example Using the Solve Order Property" on page 974). You can specify a solve order between 0 and 127. The default is 0.

You can specify the solve order at the member level or at the dimension level. Essbase uses the following information to define calculation precedence:

1. Member solve order
2. Dimension solve order (members without formulas for which you do not specify a member solve order inherit the solve order of their dimension. Members with formulas for which you do not specify a member solve order have a solve order of zero.)

   If multiple members have the same solve order, the members are evaluated in the reverse order in which their dimensions occur in the database outline. The member that occurs later in the outline takes precedence.

   The tie situation calculation order is different for calculated members defined in an MDX query for block storage databases. See the *Oracle Essbase Technical Reference*.

   Note:

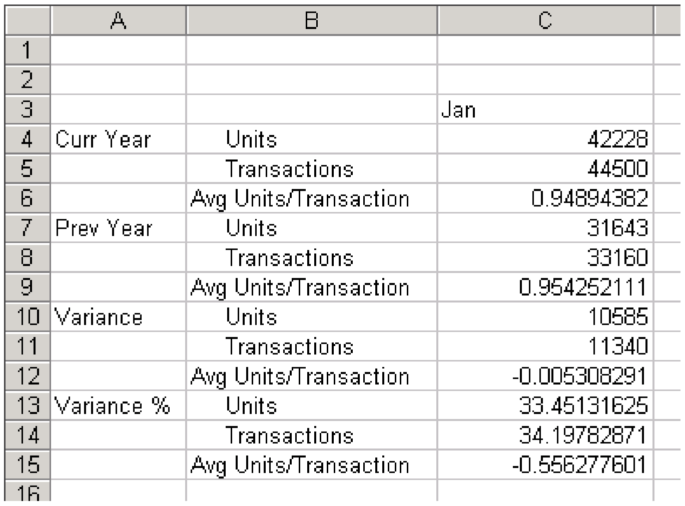
   When a member formula is dependant on the value of another member, the member with the formula must have a higher solve order than the member or members on which it depends. For example, in the ASOsamp.Sample database outline in Figure 175 on page 975, Avg Units/Transaction depends on the value of Units and of Transactions. Avg Units/Transaction must have a higher solve order than Units and Transactions.

Example Using the Solve Order Property

The following example is based on the ASOSamp.Sample database. To remove ambiguity in query results, the example uses the solve order property to specify the required calculation priority.

The spreadsheet query in Figure 174 on page 975 retrieves data for the number of units sold and the number of transactions for January of the current year and for January of the previous year. The Variance member shows the difference between the current year and the previous year. The Avg Units/Transaction member shows a ratio of the number of units sold per transaction.

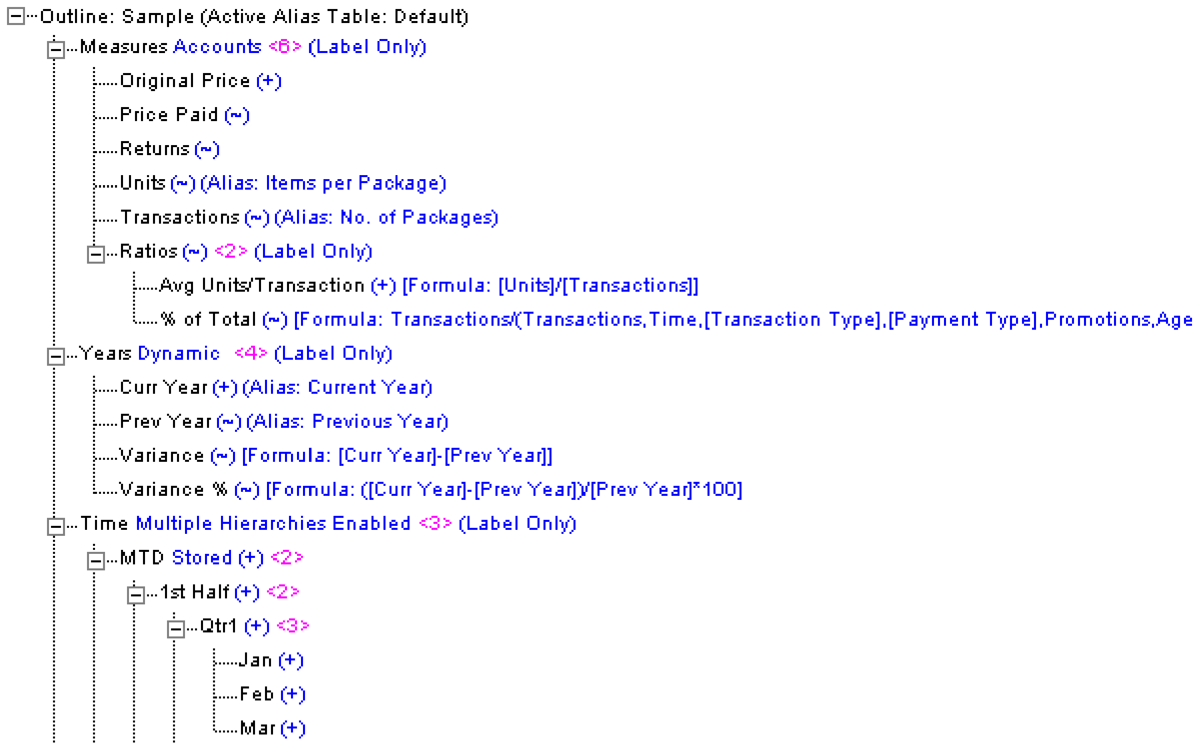
Figure 174 Results from Spreadsheet Query of ASOSamp.Sample database Showing the Variance Between Two Ratios (C12)



	A	B	C
1			
2			
3			Jan
4	Curr Year	Units	42228
5		Transactions	44500
6		Avg Units/Transaction	0.94894382
7	Prev Year	Units	31643
8		Transactions	33160
9		Avg Units/Transaction	0.954252111
10	Variance	Units	10585
11		Transactions	11340
12		Avg Units/Transaction	-0.005308291
13	Variance %	Units	33.45131625
14		Transactions	34.19782871
15		Avg Units/Transaction	-0.556277601
16			

Figure 175 shows the database outline for these members and the formulas applied to the Variance and Avg Units/Transaction members.

Figure 175 ASOSamp.Sample Database Showing the Measures, Years, and Time Dimensions



```
Outline: Sample (Active Alias Table: Default)
  Measures Accounts <6> (Label Only)
    Original Price (+)
    Price Paid (~)
    Returns (~)
    Units (~) (Alias: Items per Package)
    Transactions (~) (Alias: No. of Packages)
    Ratios (~) <2> (Label Only)
      Avg Units/Transaction (+) [Formula: [Units]/[Transactions]]
      % of Total (~) [Formula: Transactions/(Transactions,Time,[Transaction Type],[Payment Type],Promotions,Age
  Years Dynamic <4> (Label Only)
    Curr Year (+) (Alias: Current Year)
    Prev Year (~) (Alias: Previous Year)
    Variance (~) [Formula: [Curr Year]-[Prev Year]]
    Variance % (~) [Formula: ([Curr Year]-[Prev Year])/[Prev Year]*100]
  Time Multiple Hierarchies Enabled <3> (Label Only)
    MTD Stored (+) <2>
      1st Half (+) <2>
        Qtr1 (+) <3>
          Jan (+)
          Feb (+)
          Mar (+)
```

When calculating the variance of the average units per transaction (cell C12 in Figure 174 on page 975), the result could be the variance between the two ratios, or the result could be the ratio of the two variances. The result depends on whether Essbase gives precedence to the formula on Variance or the formula on Avg Units/Transaction.

The value of the solve order property, which is attached to the members in the database outline, determines the priority with which Essbase evaluates the formulas. The formula on the member that has the higher solve order takes precedence.

In the example, if the Variance member has a higher solve order than the Avg Units/Transaction member, then the formula on the Variance member takes precedence and the result is the variance between two ratios. This is the case in the ASOsamp.Sample database, because the solve order of the Variance member is 20 and the solve order of the Avg Units/Transaction member is 10. The formula on Variance takes precedence, because the Variance member has the higher solve order. The result for cell C12 of the query in Figure 174 is the variance between the two ratios, as shown in Table 137:

Table 137 Using the Solve Order Property to Specify the Variance Between Two Ratios

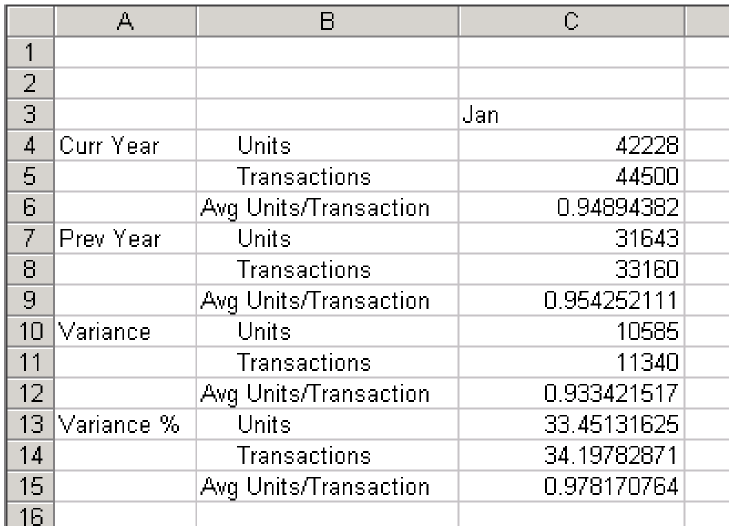
Member	Solve Order	Formula	Result of Intersection of Variance and Avg Units/ Transaction (cell C12 in Figure 174)
Variance	20	Curr Year - Prev Year	Current year average units/transaction - previous year average units/transaction
Avg Units/Transaction	10	Units/Transactions	0.94894382 (cell C6) - 0.954252111 (cell C9) = -0.005308291 (cell C12)

Alternatively, if you change the ASOsamp.Sample database, and you give the Avg Units/ Transaction member a higher solve order than the Variance member, then the formula on the Avg Units/Transaction member takes precedence, and the result is the ratio of two variances, as shown in Table 138 and in Figure 176:

Table 138 Using the Solve Order Property to Specify the Ratio of Two Variances

Member	Solve Order	Formula	Result of Intersection of Variance and Avg Units/ Transaction (cell C12 in Figure 176)
Variance	10	Curr Year - Prev Year	Variance (current year to previous year) of units / variance of transactions
Avg Units/Transaction	20	Units/Transactions	10585 (cell C10) / 11340 (cell C11) = 0.933421517 (cell C12)

Figure 176 Results from Spreadsheet Query of ASOsamp.Sample Database Showing the Ratio of Two Variances (C12)



	A	B	C
1			
2			
3			Jan
4	Curr Year	Units	42228
5		Transactions	44500
6		Avg Units/Transaction	0.94894382
7	Prev Year	Units	31643
8		Transactions	33160
9		Avg Units/Transaction	0.954252111
10	Variance	Units	10585
11		Transactions	11340
12		Avg Units/Transaction	0.933421517
13	Variance %	Units	33.45131625
14		Transactions	34.19782871
15		Avg Units/Transaction	0.978170764
16			

Aggregating an Aggregate Storage Database

Aggregate storage databases require no separate calculation step after data values are loaded into the level 0 cells of the outline. From any point in the database, users can retrieve and view values that are aggregated dynamically, for the current retrieval only. Aggregate storage databases are smaller than block storage databases, enabling quick retrieval of data values.

As databases grow, retrievals must process more data values to satisfy the calculation needs of the queries. For faster retrieval, Essbase enables you to precalculate data values and store those values in aggregations. If a database size nears one million aggregate cells, you should strongly consider performing an aggregation. Depending on database usage and the usage environment, you can achieve performance benefits by precalculating smaller databases as well. You can use either Administration Services Console or MaxL to calculate an aggregate storage database.

Understanding Aggregation-Related Terms

The following topics describe terms that are integral to an explanation of aggregate storage database calculation.

Aggregate Cells

Cells for level 0 intersections across dimensions, without formulas, are called input cells. Data values can be loaded to them. Higher-level cells involving members of the accounts dimension or dynamic hierarchies are always calculated at retrieval. All other higher-level intersections across dimensions are *aggregate cells*. For example, for the outline in Figure 171 on page 954, Price Paid > Curr Year > 1st Half > Portable Audio > CO is an aggregate cell; Original Price > Curr Year > Jan > Camcorders > CO is another aggregate cell. Values for aggregate cells must be rolled up from lower-level values.

Aggregate cell values are calculated for each request, or they can be precalculated and stored on disk.

Aggregate Views

When Essbase defines which aggregate cells to precalculate and store, it works with aggregate views. An *aggregate view* is a collection of aggregate cells. The collection is based on the levels of the members within each dimension.

For example, consider one aggregate view for the outline in Figure 171 on page 954. This aggregate view includes aggregate cells for the following dimension levels:

- Measures dimension, level 0
- Years dimension, level 0
- Time dimension, level 1 of hierarchy 0
- Product dimension, level 2 of hierarchy 0
- Geography dimensions, level 0

The example aggregate view is shown as 0, 0, 1/0, 2/0, 0.

Each dimension is shown, left to right, in its sequence in the outline. If a dimension contains hierarchies, the notation specifies the member level within its hierarchy. Hierarchies within a dimension are numbered top-down, starting with hierarchy 0.

The 0, 0, 1/0, 2/0, 0 aggregate view contains aggregate cells that include the following member intersections:

```
Original Price, Curr Year, Qtr1, Personal Electronics, CO
Original Price, Curr Year, Qtr1, Personal Electronics, KS
Original Price, Curr Year, Qtr1, Home Entertainment,   CO
Original Price, Curr Year, Qtr1, Home Entertainment,   KS
Original Price, Curr Year, Qtr2, Personal Electronics, CO
Original Price, Curr Year, Qtr2, Personal Electronics, KS
Original Price, Curr Year, Qtr2, Home Entertainment,   CO
Original Price, Curr Year, Qtr2, Home Entertainment,   KS
Original Price, Curr Year, Qtr3, Personal Electronics, CO
Original Price, Curr Year, Qtr3, Personal Electronics, KS
Original Price, Curr Year, Qtr3, Home Entertainment,   CO
Original Price, Curr Year, Qtr3, Home Entertainment,   KS
Original Price, Curr Year, Qtr4, Personal Electronics, CO
Original Price, Curr Year, Qtr4, Personal Electronics, KS
Original Price, Curr Year, Qtr4, Home Entertainment,   CO
Original Price, Curr Year, Qtr4, Home Entertainment,   KS
Original Price, Prev Year, Qtr1, Personal Electronics, CO
Original Price, Prev Year, Qtr1, Personal Electronics, KS
Original Price, Prev Year, Qtr1, Home Entertainment,   CO
Original Price, Prev Year, Qtr1, Home Entertainment,   KS
and so on...
```

Aggregations

Aggregations are consolidations, based on outline hierarchy, of level 0 data values. An aggregation contains one or more aggregate views. Essbase provides an intelligent aggregation process that selects aggregate views to be rolled up, aggregates them, and then stores the values for the cells in the selected views. If an aggregation includes aggregate cells dependent on level 0 values that are changed through a data load, the higher-level values are automatically updated at the end of the data load process.

The term *aggregation* is used for the aggregation process and the set of values stored as a result of the process.

Aggregation Scripts

Each *aggregation script* is a file that defines a particular selection of aggregate views to be materialized. The Administration Services Aggregation Design Wizard enables you to create aggregation scripts and store them in the database directory as text files with the `.csc` extension. See "Working with Aggregation Scripts" on page 985.

Performing Database Aggregations

You can use either Administration Services Aggregation Design Wizard or MaxL statements to perform an aggregation. The aggregation process has two phases:

- Aggregate view selection.
- Calculation and storage of values for the selected aggregate views. This phase is also called the materialization of the aggregation.

During the aggregate view selection phase, Essbase analyzes how calculating and storing various combinations of aggregate views might affect average query response time. As input to the analysis, you can define physical storage and performance requirements. You can also track data usage and provide the information to the analysis process. See "Selecting Views Based on Usage" on page 982.

Based on their usefulness and resource requirements, Essbase creates a list of aggregate views. Included with each view in the list is storage and performance information that applies when that aggregate view plus all other aggregate views listed above it are stored. You can choose to aggregate the listed views, select and aggregate a subset of the listed views, or rerun the selection process with different input criteria. You can also add to an aggregation the materialization of new views that are based on new selection criteria.

See "Fine-Tuning Aggregate View Selection" on page 980.

Whether or not you materialize the selection, you can save the selection of aggregate views as an aggregation script. Aggregation scripts provide flexibility and can save time because they enable you to bypass the selection process if the same selection is needed again. See "Working with Aggregation Scripts" on page 985.

After the selection process is finished, the selected aggregate views are calculated when you materialize the selected aggregate views into an aggregation.

The following process is recommended for defining and materializing aggregations:

1. After the outline is created or changed, load data values.
2. Perform the default aggregation. If desired, specify a storage stopping point.
3. Materialize the suggested aggregate views and save the default selection in an aggregation script.

4. Run the types of queries the aggregation is being designed for.
5. If query time or aggregation time is too long, consider fine-tuning the aggregation. See "Fine-Tuning Aggregate View Selection" on page 980.
6. If desired, save the aggregation selection as an aggregation script. See "Working with Aggregation Scripts" on page 985

   .

➤ To perform a database aggregation selection or materialization, use a tool:

Tool	Topic	Location
Administration Services	Calculating Aggregations to Improve Retrievals	*Oracle Essbase Administration Services Online Help*
MaxL	**execute aggregate process** **execute aggregate selection** **execute aggregate build**	*Oracle Essbase Technical Reference*

Note:

Aggregation Design Wizard enables running aggregation processes in the background. When you run an Administration Services process in the background, you can continue to use Administration Services Console for other activities at the same time as the background process is running.

Fine-Tuning Aggregate View Selection

The default selection of aggregate views proposed by Essbase provides excellent performance. However, accepting all aggregate views in the selection list does not guarantee optimum performance. For the default selection, Essbase analyzes stored hierarchies and assumes an equal chance that any aggregate cell will be retrieved. Essbase cannot account for external factors such as the amount of available memory at the time of a query. Available memory can be affected by such factors as the cache memory definition at retrieval time, or the memory other concurrent processes require.

You may want to track which data is most queried and include the results and alternate views in the aggregate view selection process. (See "Selecting Views Based on Usage" on page 982.) As you tune and test aggregations, consider the following points:

- Improving retrieval performance can increase disk storage costs and the time it takes to materialize the aggregation.
- Tracking queries may result in a set of proposed aggregate views that provide better performance for some queries than for others. Selecting proposed aggregate views can considerably improve performance time of some queries with others experiencing little improvement—but never worse—as long as query type and frequency are close to the type and frequency of queries performed during the tracking period.

- Optimizing aggregations may require an iterative, fine-tuning process.

Essbase provides information to help you select and store the right balance of aggregate views for your database. Weigh this information against what you know about your database retrieval requirements and environment. Use the following information to help you select aggregate views for an aggregation:

- The maximum storage requirement

  You can specify a storage limit for selecting aggregate views in two ways:

  - When the aggregation selection is initiated, you specify a maximum storage stopping value. Aggregate views are selected until the specified storage limit is reached or there are no more views to select.

    In Administration Services, the number you specify is the amount of storage in MB; in MaxL, the number is a factor times the size of the level 0 stored values. See the *Oracle Essbase Administration Services Online Help* and the *Oracle Essbase Technical Reference*.
  - After each analysis of the database, Essbase displays information about the level 0 input cell view followed by a list of suggested aggregate views. Displayed by each aggregate view is a storage number that includes that aggregate view and all other aggregate views it depends on. You can consider this storage number as you select the aggregate views to be included in the aggregation.
- The relative "Query Cost" performance improvement

  The Query Cost number that is displayed by each aggregate view in the list projects an average retrieval time for retrieving values from the associated aggregate view. The default view selection estimates the cost as the average of all possible queries. When using query tracking, the estimated cost is the average for all tracked queries. The cost number for a specific aggregate view can be different in different selection lists; for example, aggregate view 0, 0, 1/0, 2/0, 0 can show a different query cost in the default selection list than it would show in a selection that includes tracked queries in the analysis.

  To compute the percentage improvement, divide the query cost value for the aggregate view into the query cost value shown for storing only level 0 input cells.
- Tracked usage

  Before running an aggregate view selection, you can turn on query tracking to determine which data is retrieved most often. After some period of database activity, you can have Essbase include the usage statistics in the aggregation analysis process. See "Selecting Views Based on Usage" on page 982.
- Aggregation time

  The time it takes to perform an aggregation after the selection process completes increases for each aggregate view materialized. To determine actual aggregation time, you must perform the aggregation.

The following process is recommended for fine-tuning aggregations:

1. Perform the default aggregations described in "Performing Database Aggregations" on page 979.

2. Save the default selection in an aggregation script. See "Working with Aggregation Scripts" on page 985.
3. Turn on query tracking. See "Selecting Views Based on Usage" on page 982.
4. Have users perform their usual queries against the database or perform the batch query operations for which the aggregation is being designed. Queries from all query tools are tracked.
5. After sufficient time to capture data retrieval requirements, perform another aggregation including tracked data.
6. Analyze the proposed list of aggregate views to be stored, and select the aggregate views that you determine provide the best balance of system resources and retrieval performance.
7. Materialize the selected aggregate views and, if desired, save the selection in an aggregation script.
8. Working with aggregation scripts and various selection criteria, repeat the process until you think you have the optimum selection of aggregate views for your situation.

Note:

To optimize aggregations for different database retrieval situations, such as for generating reports or user queries, you may need to repeat the tuning process, creating an aggregation script for each situation. See "Working with Aggregation Scripts" on page 985.

Selecting Views Based on Usage

Essbase enables you to capture retrieval statistics against a database. You can then use these statistics to build aggregations tailored to retrieval patterns in your company. Also, Essbase includes alternate hierarchies in its analysis of the database when usage information is used in the aggregate view selection process.

Database usage for periodic reporting may be different than for ongoing user retrievals. To optimize different retrieval situations, consider tracking situational usage patterns and creating aggregation scripts for each situation.

Before you begin the aggregation selection process, ensure that query tracking is on and that it has been on long enough to capture representative usage.

➤ To track data query patterns, use a tool:

Tool	Topic	Location
Administration Services	Tracking Query Data for Aggregation View Selection	*Oracle Essbase Administration Services Online Help*
MaxL	**alter database**	*Oracle Essbase Technical Reference*

Query tracking holds query usage information in memory. Performing any of the following operations clears query usage information.

- Loading or clearing data
- Materializing or clearing an aggregation
- Turning off query tracking

Query tracking remains on until you turn it off, stop the application, or change the outline.

Understanding User-Defined View Selection

By default, Essbase uses internal mechanisms to decide how to create aggregations. User-defined view selection provides a way for you to influence default view selection and view selection based on query data. See "Selecting Views Based on Usage" on page 982.

Administrators may apply view selection properties to stored hierarchies to restrict Essbase from choosing certain levels for aggregation.

Note:

Secondary hierarchies are either shared or attribute hierarchies.

Property	Effect
Default	On primary hierarchies, Essbase considers all levels. It does not aggregate on secondary hierarchies unless alternative roll-ups are enabled.
Consider all levels	Considers all levels of the hierarchy as potential candidates for aggregation. This is the default for primary hierarchies, but not for secondary hierarchies.
Do not aggregate	Does not aggregate along this hierarchy. All views selected by Essbase are at the input level.
Consider bottom level only	Applies only to secondary hierarchies. Essbase considers only the bottom level of this hierarchy for aggregation.
Consider top level only	Applies only to primary hierarchies. Considers only top level of this hierarchy for aggregation.
Never aggregate to intermediate levels	Applies to primary hierarchies. Selects top and bottom levels only.

Note:

The bottom level of an attribute dimension consists of the zero-level attribute members. When a secondary hierarchy is formed using shared members, the bottom level comprises the immediate parents of the shared members.

Essbase considers only views that satisfy the selected view selection properties.

You should be familiar with the dominant query patterns of databases before changing default properties; preventing selection of certain views will make queries to those views slower while improving the speed of other queries. Similarly, enabling Consider All Levels on a secondary hierarchy may speed queries to that hierarchy while making other queries slower.

➤ To define view selection properties, use a tool:

Tool	Topic	Location
Administration Services	Setting View Selection Properties	*Oracle Essbase Administration Services Online Help*
MaxL	**execute aggregate process** and **execute aggregate selection**	*Oracle Essbase Technical Reference*

Query Hints

Query hints tell Essbase what types of queries likely will occur. For example, to indicate queries at the bottom level of the time dimension, you can specify one member at the bottom level of time, such as January. This tells Essbase that *any member* at the bottom level of time likely will be queried. Essbase gives members indicated in query hints priority when selecting views, optimizing common queries.

If no member is specified for a dimension, it means that any member of that dimension may be used in a query. For example, using a query hint of (Sales, 100, East) on Sample.Basic means that profit margin measures for level 1 products at level 1 markets is a common type of query, regardless of Year and Scenario dimensions, which were omitted.

Usage-based view selection overrides query hints. See "Selecting Views Based on Usage" on page 982. User-defined view selection overrides query hints if there is a conflict between the two. See "Understanding User-Defined View Selection" on page 983

Asterisks "*" indicate that no member is selected for a dimension.

Hint	Use Case
(Jan, *, *, *, *,)	Frequent queries to the bottom level of time.
(Qtr1, *, *, *, Cola)	Queries often include members both at the bottom level of product and the quarterly level of time.
(Jan, *, *, NY, *) (Qtr1, *, *, NY, *)	Analysis at the state level of Market tends to stay at the Quarterly level or below.

Numerous hints can exist in the same outline. Any view that fits at least one hint is given more weight than views that fit none.

Query Hints cannot contain dynamic, label-only, or shared members.

➤ To apply query hints, use a tool:

Tool	Topic	Location
Administration Services	Query Hints	*Oracle Essbase Administration Services Online Help*

Working with Aggregation Scripts

Each aggregation script represents a specific aggregate view selection against a database.

Aggregation scripts can save you time. For example, after loading new data values you need not perform another aggregate view selection. You can speed the aggregation process by using the selection stored in an aggregation script to materialize the aggregation.

Aggregation scripts also give you flexibility. You can use them to save aggregate view selections optimized for different retrieval situations; for example, you can use one script to optimize retrievals in month-end reporting and another for daily retrieval requirements.

Aggregation scripts for a database become invalid when the selection it contains is invalid for the database. Create aggregation scripts when you create aggregations. Do not manually modify aggregation script files, which may cause unpredictable results. For information about when you must create aggregations and aggregation scripts, see "Replacing Aggregations" on page 987.

Creating Aggregation Scripts

Saved aggregation scripts enable you to split up the total aggregation process. You can materialize an aggregation at a different time than when the aggregate views for the aggregation are selected. The aggregation script contains information derived during the aggregate view selection phase.

➤ To create an aggregation script, use a tool:

Tool	Topic	Location
Administration Services	Calculating Aggregations to Improve Retrievals Aggregation Design Wizard	*Oracle Essbase Administration Services Online Help*
MaxL	**query database** **execute aggregate selection**	*Oracle Essbase Technical Reference*

Aggregation scripts are stored in the database directory as text files with the `.csc` extension and are valid as long as the dimension level structure in the outline has not changed. For information about when aggregation selections are invalid, see "Replacing Aggregations" on page 987. To avoid the potential clutter of invalid aggregation script files, use the Aggregation Design Wizard or manually delete aggregation scripts when they are no longer useful.

Executing Aggregation Scripts

Executing an aggregation script materializes the aggregate views specified within it. Although you can create multiple aggregation scripts, only one aggregation can be materialized at a time.

➤ To execute an aggregation script, use a tool:

Tool	Topic	Location
Administration Services	Calculating Aggregations to Improve Retrievals Aggregation Design Wizard	*Oracle Essbase Administration Services Online Help*
MaxL	**execute aggregate build**	*Oracle Essbase Technical Reference*

Note:

When Aggregation Design Wizard displays the list of existing aggregation scripts, it lists all files with the `.csc` extension in the database directory. Only valid aggregation script files can be executed.

Optimizing Aggregation Performance

To possibly improve aggregation performance time, you can use the CALCPARALLEL configuration setting to increase the number of threads. Because each thread handles a task of building an aggregate view, the number of threads you define establishes how many aggregate views can be built concurrently. Even on single-CPU computers, increasing the CALCPARALLEL configuration setting may improve performance where building aggregates is I/O-bound, such as for databases too large to fit in memory. Because threads must share aggregate storage cache, the cache size must be sufficiently large to support the number of threads defined. Otherwise, increasing the number of threads could degrade aggregation performance time.

See the *Oracle Essbase Technical Reference*.

Clearing Aggregations

At times you might want to manually clear aggregations from the disk; for example, to make the disk space available for disk-intensive operations. Clearing aggregations clears all data, except level 0 values, from the database, releasing the disk area for other use. After aggregations are cleared, queries calculate retrieved values dynamically from level 0 values.

For information about when Essbase Server automatically clears aggregated data, see "Database restructure" in Table 123, "Outline Differences Between Aggregate Storage and Block Storage," on page 909.

➤ To clear aggregations, use a tool:

Tool	Topic	Location
Administration Services	Clearing Data from Aggregate Storage Databases	*Oracle Essbase Administration Services Online Help*
MaxL	**alter database**	*Oracle Essbase Technical Reference*

Replacing Aggregations

You can replace an aggregation by clearing the existing aggregation and materializing a different selection of aggregate views. You can perform a new aggregate view selection and materialization process, or you can run an aggregation script. Consider replacing the aggregation in the following situations:

- To fine-tune the selection of aggregate views to improve performance. See "Fine-Tuning Aggregate View Selection" on page 980.
- To create aggregations optimized for different database retrieval situations, such as for generating reports or user queries.
- To optimize an aggregation after significant growth in database size. Gradually, as the size of a database increases, an aggregation can become less efficient. Consider replacing an aggregation when performance degradation is noticeable or when the database size increases to about 150% of its original size.
- To optimize aggregations for new or different operational environments, such as memory and disk resource availability changes.

You must replace an aggregation and associated aggregation scripts after the number of levels in a dimension has been changed or one or more dimensions have been added or removed from the outline. See "Performing Database Aggregations" on page 979 and "Working with Aggregation Scripts" on page 985.

Performing Time Balance and Flow Metrics Calculations in Aggregate Storage Accounts Dimensions

The topics in this section explain how to perform time balance and flow metrics calculations on aggregate storage databases.

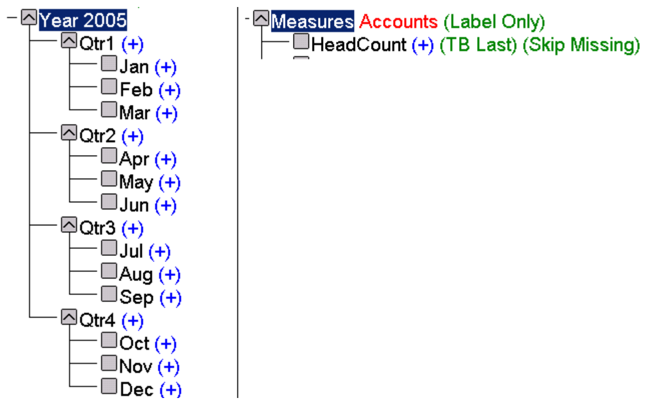
Using Time Balance Tags in Aggregate Storage Accounts Dimensions

You can set Time Balance properties on aggregate storage Accounts dimensions to provide built-in calculations along the Time dimension. This saves time and performance overhead of using member formulas to achieve time-balance functionality.

The following time-balance properties are supported on stored or formula-bearing Accounts dimension members:

- TB First, TB Last, TB Average
- SKIP NONE, SKIP MISSING

Consider a stored measure such as Headcount in a human-resources application. Within a Year-Quarter-Months hierarchy, Headcount data is loaded at the month level.



The desired yearly or quarterly Headcount value is not the sum of its months; rather, it should be the last recorded value within a time period.

Tagging Headcount as TB Last with SKIPMISSING means that, for Year 2005, its value is the last nonempty value of the headcount for its months. If Dec has a nonmissing Headcount value, then that value will be returned; otherwise, the Nov value will be checked and returned if nonmissing.

If a formula-bearing member has a time balance tag, the formula is executed only for level-0 Time members, and the Time dimension is aggregated according to the time balance tag.

The time balance tags provide a built-in calculation along the Time dimension. To perform other time-based calculations using formulas, such as period-to-date and rolling averages, you can create a dimension called TimeView and write all time-based formulas on that dimension. Doing so enables you to use Time Balance calculation functionality without losing the ability to perform other time-based calculations.

Using Flow Tags in Aggregate Storage Accounts Dimensions

A Flow tag can be assigned to Accounts dimension members bearing formulas.

The following example describes the problem to be solved with flow metrics. Assume you have Sales and Additions figures for all 12 months. You want to perform an aggregation to populate each month's beginning inventory.

Table 139 Inventory Calculation

	Sales	Additions	Inventory
Jan	5	1	50
Feb	6	3	46
Mar	4	2	43
Apr	7	0	41
...			

You would use an MDX formula on the Beginning Inventory member in order to calculate its value. Without flow metrics, to obtain each month's beginning inventory, the calculator engine would have to reiterate the MDX formula exponentially.

```
Inventory = SUM(MemberRange(Jan:Time.CurrentMember), (Additions - Sales)) +
Beg_Inventory
```

To optimize the illustrated example, assign the Inventory member the formula (**Addition – Sales**), and tag the member as Flow.

Restrictions on Alternate Hierarchies

If alternate hierarchies are used in the aggregate storage time dimension, the following restrictions apply when using Flow and TB tags on the Accounts dimension:

1. The shared level among alternate time hierarchies must be level 0.
2. The order of members at shared level among alternate time hierarchies must be the same in all alternate hierarchies.

Aggregating Time-Balance Tagged Measures

The MDX **Aggregate** function can be used to aggregate measures tagged with time balance tags. See the *Oracle Essbase Technical Reference*.

Effect of Attribute Calculations on Time Balance Measures in Aggregate Storage Databases

The following calculation logic applies if

1. The aggregate storage outline contains a time dimension or date-time dimension with one or more attribute or linked-attribute dimensions.
2. You perform queries on time balance tagged measures.

If the above cases are both true, MDX Aggregate() semantics are used to evaluate the cells.

For example, consider a scenario in which:

- Year is a time-date dimension with a day level hierarchy.
- Holiday is an attribute dimension on Year, with each date tagged with Holiday_TRUE or Holiday_FALSE.
- Opening Inventory is tagged as TBFirst.

The value of (Year, Holiday_TRUE, [Opening Inventory]) is evaluated according to the following MDX logic:

```
Aggregate( {Set of dates that are holidays in Year}, [Opening Inventory])
```

Retrieving Aggregate Storage Data

The topics in this section describe how Essbase retrieves data from aggregate storage databases.

Also see Table 127, “Query Differences Between Aggregate Storage and Block Storage,” on page 912.

Attribute Calculation Retrievals

Aggregate storage applications support only the Sum member of the Attribute Calculations dimension. If you specify any other member name from the Attribute Calculations dimension, such as Min or Avg, an error is returned. See “Understanding the Attribute Calculations Dimension” on page 174.

Retrieval Tools Supporting Aggregate Storage Databases

Essbase provides the following programs and tools that enable data retrieval from aggregate storage databases:

- MDX Queries
- Report Writer, which is run through Administration Services Console or the MaxL **export data using report_file** statement
- Spreadsheet Toolkit

Note:

Commands that support block-storage-only features (for example, the Report Writer <SPARSE command) cannot be used with aggregate storage databases. MDX queries fully support aggregate storage features.

61 Managing Aggregate Storage Applications and Databases

In This Chapter

The information in this chapter applies only to aggregate storage databases and is not relevant to block storage databases.

Also see:

- Chapter 57, "Comparison of Aggregate and Block Storage."
- *Oracle Hyperion Enterprise Performance Management System Backup and Recovery Guide* for information on backing up an aggregate storage application

Aggregate Storage Security

Defining and executing aggregations requires Calculation (Administration Services) or Execute (MaxL) permission or higher. Dimension builds that clear database values require Write permission.

Security-Related Topic	Location
"Understanding Native Security Mode in Essbase" on page 624	*Oracle Essbase Database Administrator's Guide*
"Managing User/Group Permissions for Applications and Databases"	*Oracle Essbase Administration Services Online Help*
"Privileges and Roles"	*Oracle Essbase Technical Reference*

Managing Storage for Aggregate Storage Applications

For aggregate storage applications, Tablespace Manager controls data retrieval and storage, using tablespace definitions to manage data storage and work areas on the disk.

Working with Tablespaces

Tablespaces help optimize data file and work file storage and retrieval. Tablespaces define location definitions that map data artifacts, such as aggregate views and aggregations, to files. Each application directory contains directories for four tablespaces:

- `default`—Contains database data structure and database values (After data is loaded, the tablespace location cannot be changed.)
- `log`—Contains a binary transactional log of default tablespace updates
- `metadata`—Contains a binary transactional log of default tablespace updates
- `temp`—Provides a temporary workspace to be used during operations such as data loads, aggregations, and retrievals

Tablespace names are case-sensitive, regardless of operating system. You cannot change the location or size of metadata and log. For default and temp you can specify multiple locations and sizes, and you can define tablespace properties:

- Directory path locations
- Maximum disk space to be used at each location
- Maximum file size allowed within each location

Note:

You can modify or delete file locations used to store information within a tablespace if the locations are empty.

Because Tablespace Manager allocates disk space in fixed-size increments, specifying the maximum disk space for a tablespace location specifies an end point but does not reserve the specified space.

When space is needed, Essbase checks file locations (in numerical order) and, when space is found, starts writing. When all locations are used, no space is available, and an error is returned. When database values are cleared, tablespace files shrink, releasing disk space. Work files that are no longer needed are deleted, making space available for other programs.

Based on the maximum size specified for files, Essbase writes multiple files; for example, `ess00001.dat`, `ess00002.dat`, and so on. If you back up database files to other media, do not set a maximum tablespace file size greater than the size that the media can handle.

Defining Tablespaces

You define tablespace definitions for each aggregate storage application.

➤ To define tablespaces, use a tool:

Tool	Topic	Location
Administration Services	Managing Tablespaces	*Oracle Essbase Administration Services Online Help*
MaxL	**alter tablespace**	*Oracle Essbase Technical Reference*

Note:

UNIX platforms enforce a maximum 2 GB file limit. If, during a data load or an aggregate build, the `.dat` file limit is exceeded, this message is displayed: `"Failed to extend file: file exceeds maximum file size for this system."` Essbase closes the data file, creates the next file (`ess`*`n`*`+1`), and continues.

Managing the Aggregate Storage Cache

Aggregate storage cache facilitates memory usage during data loads, aggregations, and retrievals. The cache memory locking feature is used only with block storage applications.

When an aggregate storage outline is started, a small area in memory is allocated as the aggregate storage cache for the relevant application. As additional cache area is needed, the cache size incrementally increases until the maximum cache size is used or the operating system denies additional allocations.

Note:

Denial of aggregate cache memory allocations does not deny increased use of existing memory.

You can view the current aggregate cache memory allocation and the maximum aggregate cache size setting. Changing the setting may optimize memory use. The default maximum cache size, 32 MB, is the minimum setting size. You can use the size of input-level data to determine when to increase the maximum size for the cache. Administration Services and MaxL display the size of input-level data as the aggregate storage database property: Size of level 0 values.

A 32 MB cache setting supports a database with approximately 2 GB of input-level data. If the input-level data size is greater than 2 GB by some factor, the aggregate storage cache can be increased by the square root of the factor. For example, if the input-level data size is 3 GB (2 GB * 1.5), multiply the aggregate storage cache size of 32 MB by the square root of 1.5, and set the aggregate cache size to the result: 39.04 MB.

For aggregation materialization performance, consider the number of threads set for parallel calculation. The aggregation materialization process uses multiple threads that divide the aggregate storage cache. Increasing the number of threads specified in the CALCPARALLEL configuration setting for aggregate storage applications or databases may require an increase in aggregate storage cache size. See the CALCPARALLEL configuration setting in the *Oracle Essbase Technical Reference*.

Note:

Setting the number of threads higher than the number of processors may improve aggregate storage application performance.

Do not increase the maximum size of the aggregate storage cache beyond what is needed.

➤ To set aggregate storage cache size, use a tool:

Tool	Topic	Location
Administration Services	Sizing the Aggregate Storage Cache	*Oracle Essbase Administration Services Online Help*
MaxL	**query application** **alter application**	*Oracle Essbase Technical Reference*

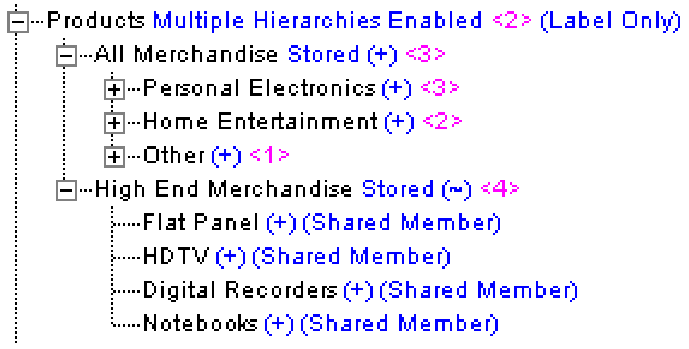
Note:

A changed aggregate storage cache setting becomes effective when the application is restarted.

Aggregate Storage Database Restructuring

Database restructures may be forced by some aggregate storage database outline changes, including changes to hierarchies. A hierarchy comprises a top member and its descendants.

- A dynamic hierarchy includes only one stored level. The Accounts dimension is a dynamic hierarchy.
- An attribute dimension is one hierarchy. The generation 1 member is the top member of the hierarchy.
- If a standard dimension is not tagged as multiple hierarchies enabled, it is one hierarchy. The generation 1 member is the top member of the hierarchy.
- If a standard dimension is tagged as multiple hierarchies enabled, it contains multiple hierarchies. The generation 2 members are the top members of the hierarchies. For example, the Products dimension in ASOSamp.Sample contains two hierarchies. The top members are the generation 2 members All Merchandise and High End Merchandise.



```
Products Multiple Hierarchies Enabled <2> (Label Only)
    All Merchandise Stored (+) <3>
        Personal Electronics (+) <3>
        Home Entertainment (+) <2>
        Other (+) <1>
    High End Merchandise Stored (~) <4>
        Flat Panel (+) (Shared Member)
        HDTV (+) (Shared Member)
        Digital Recorders (+) (Shared Member)
        Notebooks (+) (Shared Member)
```

What outlines changes affect:

- Whether data must be cleared from the database before restructuring
- The time and storage required to restructure the outline

Levels of Aggregate Storage Database Restructuring

To minimize the time and storage needed for database restructures, if a database outline changes frequently, analyze the outline and the types of outline changes.

Levels of restructuring for aggregate storage databases, listed from most to least expensive (in regard to time, storage, and data):

User—Outline Changes	**Essbase—Restructure Level**	**Performance Impact**
Add, delete, or move a standard dimension	Clears data and aggregate views. and performs full outline restructure	Very high User must reload input (level 0) data, select the aggregate views, and rerun the database aggregation.
● Add, delete, or move a hierarchy ● Change the number of stored levels in a hierarchy. See "Changes That Do and Do Not Affect the Number of Stored Levels" on page 996. ● Change the top member of a stored hierarchy from label-only to stored or from stored to label-only ● Change a dynamic hierarchy to a stored hierarchy or a stored hierarchy to a dynamic hierarchy ● Change a primary or an alternate hierarchy so that it matches or no longer matches its primary or alternate hierarchy (All level 0 members of a primary hierarchy must be represented directly or indirectly (for example, a parent that is a sum of its children may represent its children) in all alternate hierarchies. The top level of the primary hierarchy must equate to the top level of each alternate hierarchy. For an example, see "Changes in Alternate Hierarchies" on page 998.)	Clears aggregate views. and performs full outline restructure	Very high Storage requirement is up to three times the size of the database file (`.dat` file). User must select the aggregate views and rerun the database aggregation.
Perform a change that is not included in other categories; for example, delete or move a member	Performs full outline restructure	High Storage requirement is up to three times the size of the database file (`.dat` file).
Perform a light restructure change (described below) to an alternate hierarchy or an attribute dimension	Clears aggregate views that are based on attribute dimensions or alternate hierarchies, and performs a light restructure	Low User must rerun the database aggregation for aggregate views that are based on

User—Outline Changes	Essbase—Restructure Level	Performance Impact
		attribute dimensions or on alternate hierarchies. Such aggregate views exist only if you used query tracking to select views based on usage. See “Selecting Views Based on Usage” on page 982.
On nonattribute dimensions without stored level 0 members (for example, all level 0 members are shared or have formulas), add a child or child branch without changing the number of levels in the hierarchy On nonattribute dimensions with stored level 0 members: ● Add a child as the last child of a parent without crossing the 2 boundary (2, 4, 6, 8, and so on) (For example, you can add a third and fourth child but not only a third child. For an example, see “Addition of Child Members” on page 998.) ● Add a child branch as the last child branch of an existing parent without crossing the 2 boundary and without changing the number of levels in the hierarchy. (For an example, see “Addition of Child Branches” on page 999.) Examples: ● Renames a member ● Changes a formula ● Changes an alias ● Changes a dynamic hierarchy consolidation operator (for example from + to -)	Performs a light restructure; changes the outline	Very low

Outline-Change Examples

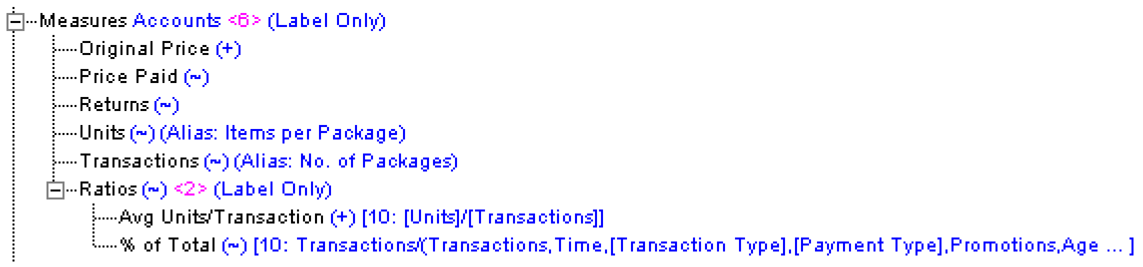
Examples of the more complicated outline changes described in the table above follow.

Changes That Do and Do Not Affect the Number of Stored Levels

Changing or not changing the number of stored levels in a hierarchy has different results.

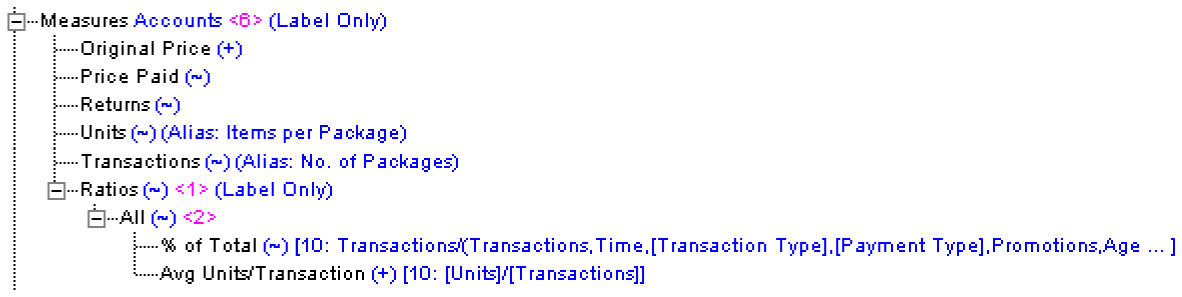
Examples That Do Not Change the Number of Stored Levels in a Hierarchy

In ASOSamp.Sample, the Measures dimension is tagged as accounts. Therefore, as a dynamic hierarchy, Measures includes only one stored level.



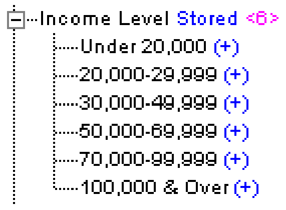
```
Measures Accounts <6> (Label Only)
    Original Price (+)
    Price Paid (~)
    Returns (~)
    Units (~) (Alias: Items per Package)
    Transactions (~) (Alias: No. of Packages)
    Ratios (~) <2> (Label Only)
        Avg Units/Transaction (+) [10: [Units]/[Transactions]]
        % of Total (~) [10: Transactions/(Transactions,Time,[Transaction Type],[Payment Type],Promotions,Age ... ]
```

Adding the child member All to Ratios does not change the number of stored levels in the Measures dimension. Saving the outline triggers a light restructure.



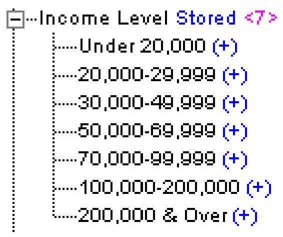
```
Measures Accounts <6> (Label Only)
    Original Price (+)
    Price Paid (~)
    Returns (~)
    Units (~) (Alias: Items per Package)
    Transactions (~) (Alias: No. of Packages)
    Ratios (~) <1> (Label Only)
        All (~) <2>
            % of Total (~) [10: Transactions/(Transactions,Time,[Transaction Type],[Payment Type],Promotions,Age ... ]
            Avg Units/Transaction (+) [10: [Units]/[Transactions]]
```

In ASOSamp.Sample, Income Level is a stored hierarchy dimension.



```
Income Level Stored <6>
    Under 20,000 (+)
    20,000-29,999 (+)
    30,000-49,999 (+)
    50,000-69,999 (+)
    70,000-99,999 (+)
    100,000 & Over (+)
```

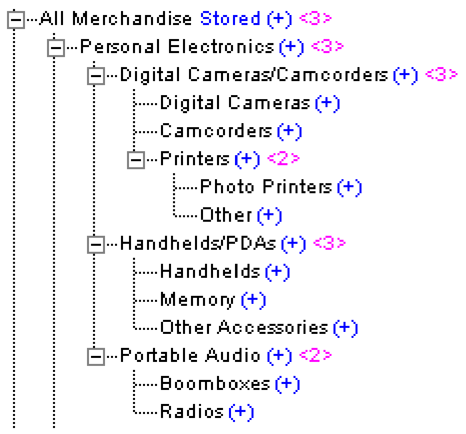
Adding a child member does not change the number of levels (two) in the hierarchy. However, adding the seventh child member crosses the 2^n boundary of 6 (see "Addition of Child Members" on page 998), requiring a full outline restructure.



```
Income Level Stored <7>
    Under 20,000 (+)
    20,000-29,999 (+)
    30,000-49,999 (+)
    50,000-69,999 (+)
    70,000-99,999 (+)
    100,000-200,000 (+)
    200,000 & Over (+)
```

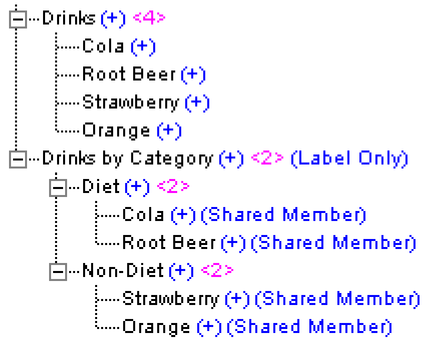
Example That Changes the Number of Stored Levels

In the Product dimension in ASOSamp.Sample, renaming Photo Printers to Printers and adding child members increases the number of levels in the All Merchandise hierarchy from four to five. When the outline is saved, Essbase clears all aggregate views and performs a full outline restructure.



```
All Merchandise Stored (+) <3>
  Personal Electronics (+) <3>
    Digital Cameras/Camcorders (+) <3>
      Digital Cameras (+)
      Camcorders (+)
      Printers (+) <2>
        Photo Printers (+)
        Other (+)
    Handhelds/PDAs (+) <3>
      Handhelds (+)
      Memory (+)
      Other Accessories (+)
    Portable Audio (+) <2>
      Boomboxes (+)
      Radios (+)
```

Changes in Alternate Hierarchies

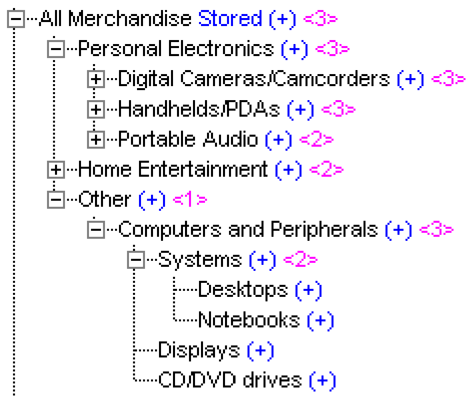


```
Drinks (+) <4>
  Cola (+)
  Root Beer (+)
  Strawberry (+)
  Orange (+)
Drinks by Category (+) <2> (Label Only)
  Diet (+) <2>
    Cola (+) (Shared Member)
    Root Beer (+) (Shared Member)
  Non-Diet (+) <2>
    Strawberry (+) (Shared Member)
    Orange (+) (Shared Member)
```

If you delete the shared member Orange under Drinks by Category and do not delete its nonshared member under Drinks, the alternate hierarchy Drinks by Category is no longer a replica of the Drinks hierarchy. When the outline is saved, Essbase clears all aggregate views and performs a full outline restructure.

If you delete the shared and nonshared Orange members, the alternate hierarchy Drinks by Category remains a replica of the Drinks hierarchy. When the outline is saved, Essbase performs a full outline restructure but does not clear aggregate views.

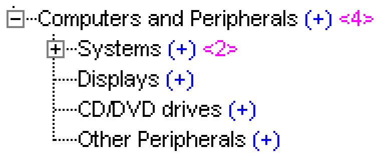
Addition of Child Members



```
All Merchandise Stored (+) <3>
  Personal Electronics (+) <3>
    Digital Cameras/Camcorders (+) <3>
    Handhelds/PDAs (+) <3>
    Portable Audio (+) <2>
  Home Entertainment (+) <2>
  Other (+) <1>
    Computers and Peripherals (+) <3>
      Systems (+) <2>
        Desktops (+)
        Notebooks (+)
      Displays (+)
      CD/DVD drives (+)
```

In ASOSamp .Sample, adding a child member under Systems in the All Merchandise hierarchy increases the number of children under Systems to three, crossing the boundary 2. When the outline is saved, Essbase clears all aggregate views and performs a full outline restructure.

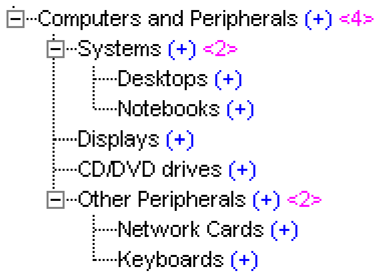
However, adding a child member under Computers and Peripherals increases the number of children under Computers and Peripherals from three to four. Adding a fourth child, which must be added after the existing members, does not cross the boundary of 2 or 4. The child must be added after existing members. When the outline is saved, Essbase performs a light restructure.



```
Computers and Peripherals (+) <4>
    Systems (+) <2>
    Displays (+)
    CD/DVD drives (+)
    Other Peripherals (+)
```

Addition of Child Branches

In ASOSamp.Sample, adding a child branch under Computers and Peripherals in the All Merchandise hierarchy increases the number of children to four. Adding a fourth child, which must be added after the existing members, does not cross the boundary of 2 or 4. The new member, called Other Peripherals, has two children. Systems (at the same level as Other Peripherals) has two children. Adding the child branch stays within the 2 boundary for children of members at the same level. When the outline is saved, Essbase performs a light restructure.



```
Computers and Peripherals (+) <4>
    Systems (+) <2>
        Desktops (+)
        Notebooks (+)
    Displays (+)
    CD/DVD drives (+)
    Other Peripherals (+) <2>
        Network Cards (+)
        Keyboards (+)
```

Adding a child branch with three child members crosses the 2 boundary and requires that Essbase clear all aggregate views and perform a full outline restructure.

Exporting Aggregate Storage Databases

If you have read permission for an aggregate storage database, you can export level 0 data from the database to a specified text file. The export file contains only uncompressed data, not control, outline, or security information. During data export, users can connect to Essbase Server and perform read-only operations on the database.

Exported data can be reloaded without a rules file if there are no outline changes. Consider exporting data for the following reasons:

- To transfer data across platforms
- To create an exported file in text, rather than binary, format
- To create backups

Exports

The default location for export files is *ARBORPATH*/app/. You can specify an alternate location; see the *Oracle Essbase Administration Services Online Help* or the *Oracle Essbase Technical Reference*.

Aggregate storage database exports have limits:

- You can export only level 0 data (input data).
- You cannot perform columnar exports. In a columnar export, the output file displays a member name from each dimension in every row (and names can be repeated from row to row).

To avoid creating export files larger than 2 GB, Essbase may create multiple export files that include a number suffix in the name, as follows: _1, _2, and so on. For example, if the first file name is `/home/exportfile.txt`, the next file is `/home/exportfile_1.txt`.

To improve performance, you can export data in parallel.

➤ To export data, use a tool:

Tool	Topic	Location
Administration Services	Exporting Databases	*Oracle Essbase Administration Services Online Help*
MaxL	**export data**	*Oracle Essbase Technical Reference*

Part XI

Appendices

In Appendices:

A Limits

In This Appendix

In this appendix, the term non-Unicode applications refers to non-Unicode-mode applications and applications on Essbase Servers that are not Unicode-enabled. See Chapter 41, "Understanding the Essbase Unicode Implementation."

Names and Related Artifacts

Table 140 lists limits for names and related artifacts.

Table 140 Names and Related Artifacts

Artifact	Limits
Alias name	● Non-Unicode application limit: 80 bytes ● Unicode-mode application limit: 80 characters
Alias table name	● Non-Unicode application limit: 30 bytes ● Unicode-mode application limit: 30 characters
Application name	● Non-Unicode application limit: 8 bytes ● Unicode-mode application limit: 30 characters
Application description	● Non-Unicode application limit: 79 bytes ● Unicode-mode application limit: 80 characters
● Custom-defined function name ● Custom-defined macro name ● Custom-defined function specification ● Custom-defined macro specification	● Non-Unicode application limit: 127 bytes. MaxL and the API truncate characters after 127 bytes. ● Unicode-mode application limit: 128 characters. MaxL and the API truncate characters after 128 characters. In either case, no truncation on server. No error is displayed if truncation occurs.

Artifact	Limits
Custom-defined function and macro comment	● Non-Unicode application limit: 255 bytes. After 255 bytes, characters are truncated by MaxL and API. ● Unicode-mode application limit: 256 characters. MaxL and the API truncate characters after 256 characters. In either case, no truncation on server. No error is displayed if truncation occurs.
Database name	● Non-Unicode application limit: 8 bytes ● Unicode-mode application limit: 30 characters
Database description	● Non-Unicode application limit: 79 bytes ● Unicode-mode application limit: 80 characters
Directory path For example: `Hyperion/products/Essbase/EssbaseServer/bin`	● Non-Unicode application limit: 256 bytes ● Unicode-mode application limit: 1024 bytes
Environment variable name	320 bytes (Unicode- and Non-Unicode-mode applications)
Environment variable value	256 bytes (Unicode- and Non-Unicode-mode applications)
File name for calculation scripts, report scripts, or rules files	● Non-Unicode application limit: 8 bytes ● Unicode-mode application limit: If included within a path, the smaller of the following two values: ❍ 1024 bytes ❍ The limit established by the operating system If not included within a path, as in some MaxL statements, 1024 bytes.
Filter name	● Non-Unicode application limit: 30 bytes ● Unicode-mode application limit: 30 characters
Group name	● Non-Unicode application limit: 30 bytes ● Unicode-mode application limit: 30 characters
LRO cell note	● Non-Unicode application limit: 600 bytes ● Unicode-mode application limit: 600 characters
LRO URL	512 characters (always single-byte characters)
LRO description for URLs and files	● Non-Unicode application limit: 80 bytes ● Unicode-mode application limit: 80 characters
Member comment field	● Non-Unicode application limit: 255 bytes ● Unicode-mode application limit: 256 characters
Member name	● Non-Unicode application limit: 80 bytes

Artifact	Limits
	● Unicode-mode application limit: 80 characters
Essbase Server name	● Non-Unicode application limit: 29 bytes ● Unicode-mode application limit: 50 characters
Qualified member name limit: the number of levels that can be specified in a qualified member name	● 20 levels
Password	● Non-Unicode application limit: 100 bytes ● Unicode-mode application limit: 100 characters
Substitution variable name	320 bytes
Substitution variable value	256 bytes
Trigger name	● Non-Unicode application limit: 30 bytes ● Unicode-mode application limit: 30 characters
UDA	● Non-Unicode application limit: 80 bytes ● Unicode-mode application limit: 80 characters
User name	● Non-Unicode application limit: 30 bytes ● Unicode-mode application limit: 30 characters

Data Load and Dimension Build Limits

Table 141 lists limits related to data loading and dimension building.

Table 141 Data Load and Dimension Build Limits

Artifact	Limits
Selection and rejection criteria	Number of characters that describe selection and rejection criteria: combination of all criteria limited to 32 KB
Number of error messages written to a data load or dimension build error log (DATAERRORLIMIT in `essbase.cfg`)	Default 1000, minimum 1, maximum 65000

Aggregate Storage Database Limits

Table 142 lists limits related to aggregate storage databases.

Table 142 Aggregate Storage Limits

Artifact	**Limits**
Number of hierarchies in a single dimension of an aggregate storage database	256 (including all stored hierarchies, dynamic hierarchies, and any attribute dimensions based on the dimension)
Number of members in an aggregate storage outline	10,000,000 to 20,000,000 members, depending on available memory and other memory requirements
Maximum file location size in an aggregate storage database	4,294,967,295 MB
Number of stored dimension level combinations in an aggregate storage outline	The product of dimension level factors across stored dimensions cannot exceed 2^{52}, which is equal to 4,503,599,627,370,496. To calculate the number of stored dimension levels in an outline: 1. Determine the stored-level factor for each dimension. 2. Multiply the dimension factors together; for example, *a* * *b* * *c* * *d*, and so on. To determine the factor for each dimension, use the following guidelines: ● For dynamic dimensions, the factor is 1. ● For stored dimensions, for an initial dimension factor, count the nonlabel-only levels. ❍ If a stored dimension has attribute dimensions, count the levels in each attribute dimension, and add that number to the dimension factor. Then, if all level 0 members of the dimension have an attribute from a specific attribute dimension, subtract 1 for each attribute dimension in which this occurs. ❍ If a stored dimension has additional stored hierarchies, count the number of nonlabel-only members in each stored hierarchy, and add the count to the dimension factor. Do not count a level if it comprises only shared members. ❍ Dynamic hierarchies in a stored dimension do not affect the dimension factor. For example, multiplying the factors for the 11 dimensions in ASOsamp.Sample, 1 * 1 * 4 * 2 * 2 * 2 * 3 * 2 * 6 * 7 * 7 equals 56,448.
Maximum number of cells that can be queried in an aggregate storage database that has a very sparse data set	2^{64} The limit is based on the number of cells that Essbase processes in the query, which is the product of the member count across all dimensions, not on the number of cells contained in the output report. You might encounter this limit when using a client interface (such as Oracle Hyperion Financial Reporting, Fusion Edition) that suppresses missing data in order to generate a relatively small report compared to the size of the query.

Block Storage Database Limits

Table 143 lists limits related to block storage databases.

Table 143 Block Storage Database Limits

Artifact	Limits
Number of members in a block storage outline	Approximately 1,000,000 explicitly defined in an Essbase outline. Hybrid Analysis and some uses of partitions enable access to many more members than are explicitly listed in an outline; the actual number of members accessible through the database is much greater. Longer names, such as can occur if multibyte characters are used, decrease the number of members that are allowed.
Number of possible blocks in a block storage database	The product of the stored member count of sparse dimensions cannot exceed $2^{104.}$ For instructions on how to calculate the number of possible blocks in a block storage database, see "Potential Number of Data Blocks" on page 1021.

Other Size or Quantity Limits

Table 144 lists other size and quantity limits.

Table 144 Other Size or Quantity Limits

Artifact	Limits
Caches: data, data file, and index	2 GB
Formula size	64 KB
Number of security filters	● Per Essbase Server: 65535 ● Per Essbase database: 32290
Number of users and groups (combined)	30,000 Exceeding this limit results in an error.
Number of aggregate views within an aggregation	1023
Rules file maximum record size	64 KB

B Setting Up Sample Applications

In This Appendix

Introduction

The Essbase Server installation includes sample applications that are designed to familiarize you with Essbase features. Each application contains one or more databases and each database has an associated data load file, as described in Table 145:

Table 145 Sample Databases and Data Load Files

Application	Database	Data Load File
Sample Interntl and Xchgrate databases demonstrate Essbase Currency Conversion features.	Basic Interntl Xchgrate	Calcdat.txt Currcalc.txt Rates.txt
Demo See the *Oracle Essbase Technical Reference* and documentation for Essbase Report Writer.	Basic	Data.txt
Samppart Demonstrates Essbase Partitioning features. See Chapter 14, "Designing Partitioned Applications."	Company	Calccomp.txt
Sampeast Demonstrates Essbase Partitioning features. See Chapter 14, "Designing Partitioned Applications."	East	Calceast.txt
Sample_U Contains a Unicode-mode version of the Basic database in the Sample application, which includes alias tables in English and four other character sets. Characters in this application are encoded in UTF-8.	Basic	Calcdat.txt

Application	Database	Data Load File
DMDemo[*] Demonstrates Data Mining features.	Basic	dmdemo.txt
ASOsamp Demonstrates aggregate storage features.	Sample	dataload.txt[†]

[*]The DMDemo sample application uses the DM_APP application. Do not remove the *ARBORPATH*/`app`/`$DM_APP$` folder or any files in the folder.

[†]The ASOSamp.Sample database must be loaded using the `dataload.rul` rules file. The other samples do not require a rules file.

Loading Data into Sample Databases

Before you can use a sample application, you must load data into it.

Note:

Before loading data into the Partitioning sample databases, first create a partition user or other username. See "Creating the Partition User" on page 1012.

➤ To load data into a sample database:

1 **Open the Administration Services Console.**

2 **From Enterprise View or a custom view, navigate to the appropriate Essbase Server, application, and database.**

3 **Right-click the database and select Load data.**

4 **Set the Data Load options.**

- For Data Source Type, select **Data file.**
- For Mode, choose one of the following options:
  - Load only: Performs a data load.
  - Build only: Performs a dimension build.
  - Both: Performs both a data load and a dimension build.
-
- For Data Source, click **Find Data File**. In the Essbase Server tab, navigate to the correct data source. For **Files of Type**, ensure that **Data files** (***.txt**) is selected.
- For Rules File, if the data source requires a rules file, click **Find Rule File**. In the **Essbase Server** tab, navigate to the rule file. For **Files of Type**, ensure that **Rules file** (***.rul**) is selected.

  Note:

  The ASOsamp database must be loaded using the `dataload.rul` file. The other samples do not require a rules file.

When the data load is completed, a dialog box appears with information about the files that loaded and whether there were errors.

Providing User Access to Sample Applications

Essbase provides a comprehensive security system for a secure multiple-user environment. By default, the sample applications are created with a security access level of None, which means that no user can connect to the sample databases unless the user is defined as an administrator.

The system administrator, defined when installing Essbase, automatically holds administrator privileges. Therefore, the system administrator can make the sample applications available to other users.

➤ To provide all users with Write access to a sample application:

1. **Log on to Administration Services Console using the system administrator account.**
2. **From Enterprise View, find the appropriate Essbase Server and application.**
3. **Right-click the application and select Edit properties.**
4. **In Application Properties, select the General tab.**
5. **For Minimum access level, select Write.**

   For example, if you want all users to have at least Write access to all databases in the application (meaning that all users can update data values), select Write.
6. **From the Minimum Database Access group, select Write.**
7. **Click Apply.**
8. **Repeat this procedure for each database.**

   The selected application is ready for use. If you want to provide access to another application, repeat the procedure.

Preparing the Aggregate Storage Sample Application

➤ To prepare the sample aggregate storage application for use:

1. **Load data using the following data and rules files:**

   ```
   ARBORPATH/app/ASOsamp/Sample/dataload.txt
   ARBORPATH/app/ASOsamp/Sample/dataload.rul
   ```

   For instructions on loading data to aggregate storage databases, see the *Oracle Essbase Administration Services Online Help*.
2. **Precalculate aggregations on the database to improve retrieval times, using one of these methods:**
   - Aggregation Design Wizard in Administration Services.

     See the *Oracle Essbase Administration Services Online Help*.

- **execute aggregate process** MaxL statement.

  See the *Oracle Essbase Technical Reference*.

3 **Make sure that the `essbase.jar` file is in the *`ESSBASEPATH`*`/java` directory.**

Preparing the Partitioning Sample Applications

Setting the Environment

Essbase includes two sample applications and databases that demonstrate the features of Partitioning:

- Samppart—Company
- Sampeast—East

The Partitioning applications and databases include partition definitions stored in `.ddb` files. The `.ddb` files define the map between member combinations in the target database, Company, and the source database, East.

For the Partitioning applications to work in your environment, you may need to create a user named *partitionuser* (see "Creating the Partition User" on page 1012), or change the embedded user names in the `.ddb` files (see "Changing Embedded User Names in Sample Partition Definitions" on page 1013).

Note:

Do not make changes to username information directly in the `.ddb` files.

Creating the Partition User

Before you work with the Samppart and Sampeast applications, you may need to create a user named `partitionuser`, which must have Application Manager access to both applications.

➤ To create `partitionuser`:

1 **Log on to Administration Services Console using the system administrator account.**

2 **In Enterprise View, navigate to the Administration Server's node and select the appropriate Essbase Administration Server name.**

3 **Right-click the User node and select Create user.**

4 **For Create user on Essbase Administration Server, enter `partitionuser` for Username.**

5 **For Password, type a password.**

6 **For Confirm Password, type the password again.**

7 **For Administrator privileges, select true.**

8 **Click OK.**

Changing Embedded User Names in Sample Partition Definitions

If you choose not to create `partitionuser`, you can change the embedded username in the partition definition files (`.ddb`) to the username of your choice, as long as that user has administrator privileges. The `.ddb` files shipped with Samppart and Sampeast are based on the server name "localhost."

➤ To change the username in the Samppart.Company and Sampeast.East `.ddb` files:

1. Open Administration Services Console.
2. In Enterprise View, expand the **Applications** node and select the **Samppart** application.
3. Expand the **Databases** node and select the **Company** database.
4. Expand the **Partitions** node, select **Source Databases**, and then double-click:

   ```
   servername:Sampeast:East [transparent]
   ```
5. For **Data Source** and **Data Target**, select a username from **User** list.
6. Enter the password.
7. Click **Repair** to save your changes.

Note:

If, in the Repair Partition dialog box, you changed the username for the Data Source and the Data Target groups, you do not need to repeat this process to change the username in the Sampeast.East `.ddb` file.

C

Troubleshooting Essbase Errors

In This Appendix

Understanding Fatal Error Handling

The Essbase Server Kernel considers the following errors fatal:

- One or more control fields have unexpected or inconsistent values.
- The kernel detects data corruption.
- The kernel cannot perform an operation that is necessary to ensure data integrity (for example, disk space is insufficient).
- The kernel encounters a condition that can lead to data corruption.

When the kernel encounters a fatal error, it shuts down and restarts, attempting to reinitialize itself and proceed with database recovery. When recovery begins, Essbase displays an error message similar to this one:

```
1080022 Reinitializing the Essbase Kernel for database dbname due to a
fatal error...
```

This message is followed by other informational messages related to database recovery, such as this one:

```
1080028 Performing transaction recovery for database dbname during fatal
error processing.
```

When you see such messages, the kernel has shut down and is attempting to restart. Check the Essbase Server log and determine whether Essbase issued a fatal error message just before it generated the reinitialization messages. See "Using Essbase Server and Application Logs" on page 740.

If the kernel did encounter a fatal error, you must usually restart any operation that was active at the time of the fatal error. If the operation was a calculation or a data load, you may be able to continue where the operation left off; check the Essbase Server log to see how far Essbase processed the operation. When in doubt, restart the operation. See "What to Expect if a Server Interruption Occurs" on page 791.

If the kernel did not encounter a fatal error, contact your software provider's technical support to determine what caused the kernel to shut down and restart.

See "Contents of the Essbase Server Log" on page 731 for descriptions of the contents of the Essbase Server log. See "Essbase Server and Application Log Message Categories" on page 738 for information about identifying the component where the error occurred.

Recovering from Dense Restructure Failures

If an error or system failure occurs while Essbase is restructuring, it probably will occur during step 2 in the procedure "Dense Restructures" on page 843.

➤ To recover from a failure during step 2 of "Dense Restructures" on page 843:

1. **Delete the temporary files, to free disk space and to avoid conflicts during the next database restructure.**
2. **Restart the database.**

➤ To recover from a failure during step 1, step 3, or step 4 of "Dense Restructures" on page 843:

1. **Review the disk directory and determine how far restructuring has progressed.**
2. **If all but step 4 is complete, rename the temporary files to the correct file names.**

Recovering from Sparse Restructure Failure

If a system failure occurs during any step of a sparse restructure, you can recover by restarting the database.

Synchronizing Member Names in Report Scripts and Database Outlines

When you run a report, ensure that the member names in the report script match the member names in the database outline. An error displays whenever the Report Extractor cannot find a matching member name; you must correct the name in the report script before the report continues processing.

Handling Essbase Server Problems When Running Multiple Reports

Your server computer may freeze if you try to run more reports in parallel than you have assigned server thread resources.

If you are running multiple report scripts and your server freezes, check the value of the configuration file setting SERVERTHREADS in the `essbase.cfg` file on the server. At least one thread should exist for each report running. For example, if you are running 22 reports, the value for SERVERTHREADS should be at least 22.

References

Chapters related to partitions, currency conversion, and data load have basic troubleshooting information in the relevant chapters.

For information about restoring from backups, see the *Oracle Hyperion Enterprise Performance Management System Backup and Recovery Guide*.

For information about specific error messages, see *Oracle Essbase Error Message Reference*.

For information about error and exception logs, see "Using Essbase Logs" on page 729.

D Estimating Disk and Memory Requirements

In This Appendix

Introduction

This appendix uses a worksheet approach to help you keep track of the many components that you calculate. If you are using a printed version of this book, you can photocopy the worksheets. Otherwise, you can simulate the worksheets on your own paper. Labels, such as DA and MA, help you keep track of the various calculated disk and memory component values.

How Essbase Stores Data

To size a database, you must understand the units of storage that Essbase uses. This discussion assumes that you are familiar with the following basic concepts:

- How Essbase stores data, as described in Chapter 3, “Understanding Multidimensional Databases”
- How Essbase Kernel components manage Essbase data, as described in Chapter 46, “Managing Database Settings”

An Essbase database comprises many components. In addition to an outline file and data file, Essbase uses several types of files and memory structures to manage data storage, calculation, and retrieval operations.

Table 146 describes the major components to consider when you estimate the disk and memory requirements of a database. “Yes” means the type of storage indicated is relevant, “No” means that it is not.

Table 146 Storage Units Relevant to Calculation of Disk and Memory Requirements

Storage Unit	Description	Disk	Memory
Outline	A structure that defines all elements of a database. The number of members in an outline determines the outline size.	Yes	Yes

Storage Unit	Description	Disk	Memory
Data files	Files in which Essbase stores data values in data blocks in data files. Named `essxxxxx.pag`, where *xxxxx* is a number. Essbase increments the number, starting with `ess00001.pag`, on each disk volume. Memory is also affected, because Essbase copies the files into memory.	Yes	Yes
Data blocks	Subdivisions of a data file. Each block is a multidimensional array that represents all cells of all dense dimensions relative to a particular intersection of sparse dimensions.	Yes	Yes
Index files	Files that Essbase uses to retrieve data blocks from data files. Named `essxxxxx.ind`, where *xxxxx* is a number. Essbase increments the number, starting with `ess00001.ind`, on each disk volume.	Yes	Yes
Index pages	Subdivisions of an index file. Contain index entries that point to data blocks. The size of index pages is fixed at 8 KB.	Yes	Yes
Index cache	A buffer in memory that holds index pages. Essbase allocates memory to the index cache at startup of the database.	No	Yes
Data file cache	A buffer in memory that holds data files. When direct I/O is used, Essbase allocates memory to the data file cache during data load, calculation, and retrieval operations, as needed. Not used with buffered I/O.	No	Yes
Data cache	A buffer in memory that holds data blocks. Essbase allocates memory to the data cache during data load, calculation, and retrieval operations, as needed.	No	Yes
Calculator cache	A buffer in memory that Essbase uses to create and track data blocks during calculation operations.	No	Yes

Determining Disk Space Requirements

Essbase uses disk space for its server software and for each database. Before estimating disk storage requirements for a database, you must know how many dimensions the database includes, the sparsity and density of the dimensions, the number of members in each dimension, and how many of the members are stored members.

➤ To calculate the disk space required for a database:

1 **Calculate the factors identified in “Calculating the Factors to Be Used in Sizing Disk Requirements” on page 1021.**

2 **Use the process described in “Estimating Disk Space Requirements for One Database” on page 1025 to calculate the space required for each component of a single database. If a server contains more than one database, you must perform calculations for each database.**

3 **Use the procedure outlined in “Estimating the Total Essbase Server Disk Space Requirement” on page 1032 to calculate the final estimate for the server.**

Note:

The database sizing calculations in this chapter assume an ideal scenario with an optimum database design and unlimited disk space. The space required is difficult to determine precisely because most multidimensional applications are sparse.

Calculating the Factors to Be Used in Sizing Disk Requirements

Before estimating disk space requirements for a database, you must calculate the factors to be used in calculating the estimate. Later in the chapter, you use these values to calculate the components of a database. For each database, you add together the sizes of its components.

Table 147 lists the sections that provide instructions to calculate these factors. Go to the section indicated, perform the calculation, then write the calculated value in the Value column.

Table 147 Factors Affecting Disk Space Requirements of a Database

Database Sizing Factor	Label	Value
"Potential Number of Data Blocks" on page 1021	DA	
"Number of Existing Data Blocks" on page 1022	DB	
"Size of Expanded Data Block" on page 1023	DC	
"Size of Compressed Data Block" on page 1024	DD	

Potential Number of Data Blocks

The potential number of data blocks is the maximum number of data blocks possible in the database.

If the database is already loaded, you can see the potential number of blocks on the Statistics tab of the Database Properties dialog box of Administration Services.

If the database is not already loaded, you must calculate the value.

➤ To determine the potential number of data blocks, assume that data values exist for all combinations of stored members.

1 **Using Table 148 on page 1022 as a worksheet, list each sparse dimension and its number of stored members. If more than seven sparse dimensions exist, list the dimensions elsewhere and include all sparse dimensions in the calculation.**

The following types of members are not stored members:

- Members from attribute dimensions
- Shared members
- Label Only members
- Dynamic Calc members (Dynamic Calc And Store members are stored members)

2 **Multiply the number of stored members of the first sparse dimension (line a.) by the number of stored members of the second sparse dimension (line b.) by the number of stored members of the third sparse dimension (line c.), and so on. Write the resulting value to the cell labeled DA in Table 147 on page 1021.**

```
a * b * c * d * e * f * g (and so on)
= potential number of blocks
```

Table 148 List of Sparse Dimensions with Numbers of Stored Members

Enter Sparse Dimension Name	Enter Number of Stored Members
	a.
	b.
	c.
	d.
	e.
	f.
	g.

Example

The Sample.Basic database contains the following sparse dimensions:

- Product (19 stored members)
- Market (25 stored members)

Therefore, there are 19 * 25 = 475 potential data blocks.

Number of Existing Data Blocks

As compared with the potential number of blocks, *existing blocks* refers to those data blocks that Essbase actually creates. For Essbase to create a block, at least one value must exist for a combination of stored members from sparse dimensions. Because many combinations can be missing, the number of existing data blocks is usually much less than the potential number of data blocks.

➤ To see the number of existing blocks for a database that is already loaded, look for the number of existing blocks on the Statistics tab of the Database Properties dialog box of Administration Services. Write the value in the cell labeled DB in Table 147 on page 1021.

If the database is not already loaded, you must estimate a value.

➤ To estimate the number of existing data blocks:

1 **Estimate a database density factor that represents the percentage of sparse dimension stored-member combinations that have values.**

2 **Multiply this percentage against the potential number of data blocks and write the number of actual blocks to the cell labeled DB in Table 147 on page 1021.**

```
number of existing blocks
= estimated density
* potential number of blocks
```

Example

The following three examples show different levels of sparsity and assume 100,000,000 potential data blocks:

- Extremely sparse: 5 percent of potential data cells exist.

  ```
  .05 (estimated density)
  * 100,000,000 (potential blocks)
  = 5,000,000 existing blocks
  ```

- Sparse: 15 percent of potential data cells exist.

  ```
  .15 (estimated density)
  * 100,000,000 (potential blocks)
  = 15,000,000 existing blocks
  ```

- Dense: 50 percent of potential data cells exist.

  ```
  .50 (estimated density)
  * 100,000,000 (potential blocks)
  = 50,000,000 existing blocks
  ```

Size of Expanded Data Block

The potential, expanded (uncompressed) size of each data block is based on the number of cells in a block and the number of bytes used for each cell. The number of cells is based on the number of stored members in the dense dimensions. Essbase uses eight bytes to store each intersecting value in a block.

➤ To see the number of existing blocks for a database that is loaded, look for the size of an expanded data block on the Statistics tab of the Database Properties dialog box of Administration Services.

If the database is not loaded, you must estimate the value.

➤ To determine the size of an expanded data block:

1 **Using Table 149 on page 1024 as a worksheet, enter each dense dimension and its number of stored members. If more than seven dense dimensions exist , list the dimensions elsewhere, and include all dense dimensions in the calculation.**

The following types of members are not stored members:

- Members from attribute dimensions
- Shared members
- Label Only members
- Dynamic Calc members (Dynamic Calc and Store members are stored members)

2 Multiply the number of stored members of the first dense dimension (line a) by the number of stored members of the second dense dimension (line b) by the number of stored members of the third dense dimension (line c), and so on, to determine the total number of cells in a block.

```
a * b * c * d * e * f * g (and so on)
= the total number of cells
```

3 Multiply the resulting number of cells by 8 bytes to determine the expanded block size. Write the resulting value to the cell labeled DC in Table 147 on page 1021.

```
(Total number of cells) * 8 bytes per cell
= expanded block size
```

Table 149 Determining the Size of a Data Block

Enter Dense Dimension Name	Number of Stored Members
	a.
	b.
	c.
	d.
	e.
	f.
	g.

Example

The Sample.Basic database contains the following dense dimensions:

- Year (12 stored members)
- Measures (8 stored members)
- Scenario (2 stored members)

Perform the following calculations to determine the potential size of a data block in Sample.Basic:

```
12 * 8 * 2 = 192 data cells

192 data cells
* 8 bytes
= 1,536 bytes (potential data block size)
```

Size of Compressed Data Block

Compression affects the actual disk space used by a data file. The four types of compression (bitmap, run-length encoding (RLE), zlib, and index-value) affect disk space differently. For a discussion of data compression unrelated to estimating size requirements, see "Data Compression" on page 772.

If you are not using compression, or if you have enabled RLE compression, skip this calculation and proceed to "Stored Data Files" on page 1026.

Note:

Due to sparsity also existing in the block, actual (compressed) block density varies widely from block to block. The calculations in this discussion are only for estimation purposes.

➤ To calculate an average compressed block size when bitmap compression is enabled:

1 **Determine an average block density value.**

- If the database is loaded, you can see the size of an expanded data block on the Statistics tab of the Database Properties dialog box of Administration Services. Use the value that is displayed for Block Density.
- If you want to estimate block density prior to loading data, estimate the ratio of existing data values to potential data values.

2 **To determine the compressed block size, perform the following calculation and write the resulting block size to the cell labeled DD in Table 147 on page 1021.**

```
expanded block size * block density
= compressed block size
```

Example

Assume an expanded block size of 1,536 bytes and a block density of 25% (.25):

```
1,536 bytes
* .25
= 384 bytes (compressed block size)
```

Estimating Disk Space Requirements for One Database

To estimate the disk-space requirement for a database, make a copy of Table 150 or use a separate sheet of paper as a worksheet for one database. If multiple databases are on a server, repeat this process for each database. Write the name of the database on the worksheet.

Each row of this worksheet refers to a section that describes how to size that component. Perform each calculation and write the results in the appropriate cell in the Size column. The calculations use the factors that you wrote in Table 147 on page 1021.

Table 150 Worksheet for Estimating Disk Requirements for a Database

Database Name:	
Database Component	**Size**
"Stored Data Files" on page 1026	DE
"Index Files" on page 1028	DF
"Fragmentation Allowance" on page 1029	DG
"Outline" on page 1029	DH
"Work Areas" on page 1031 (sum of DE through DH)	DI

Database Name:	
Database Component	**Size**
"Linked Reporting Objects Considerations" on page 1031, if needed	DJ
Total disk space required for the database. Total the size values from DE through DJ and write the result to Table 151 on page 1033.	

After writing all the sizes in the Size column, add them together to determine the disk space requirement for the database. Add the database name and size to the list in Table 151 on page 1033. Table 151 is a worksheet for determining the disk space requirement for all databases on the server.

Repeat this exercise for each database on the server. After estimating disk space for all databases on the server, proceed to "Estimating the Total Essbase Server Disk Space Requirement" on page 1032.

The following sections describe the calculations to use to estimate components that affect the disk-space requirements of a database.

Stored Data Files

The size of the stored database depends on whether the database is compressed and the compression method chosen for the database. Essbase provides five compression-method options: bitmap, run-length encoding (RLE), zlib, index-value, and none.

Calculating the size of a compressed database is complex for a number of reasons including the following:

- The compression method used can vary by block.
- In some situations Essbase chooses what it considers the best method at the block level.

For a comprehensive discussion of data compression unrelated to estimating size requirements, see "Data Compression" on page 772. The calculations in this discussion are for estimation purposes only.

The calculation for the space required to store the compressed data files (`ess`*`xxxxx`*`.pag`) uses the following factors:

- Number of existing blocks (value DB from Table 147 on page 1021)
- Fixed-size overhead (72 bytes per block)
- Size of expanded data block (value DC from Table 147 on page 1021)
- Database density: the percentage of sparse dimension stored-member combinations that have values. (To calculate, see "Number of Existing Data Blocks" on page 1022.)

Calculations for No Compression

➤ To calculate database size when the compression option is none, use the formula:

```
Number of blocks * (72 bytes + size of expanded data block)
```

Write the result in cell labeled DE in Table 150 on page 1025. Proceed to “Index Files” on page 1028.

Calculations for Compressed Databases

Because the compression method used can vary per block, the following calculation formulas are general estimates of the database size. Actual implementation could result in numbers larger or smaller than the calculations.

Bitmap Compression

➤ To estimate database size when the compression option is bitmap, use the formula:

```
Number of blocks
* (72 bytes
+ (average size in bytes of an expanded data block)
* (1/64 + proportion of cells that are not #Missing))
```

For example, if there are 1000 blocks, the average expanded block size is 2000 bytes, and 30% of cells are #missing, the result is:

```
1000 * (72 + 2000 * (1/64 + 0.7) )
= 1000 * 1503
= 1,503,000 bytes (approximately)
```

-

Write the result in cell labeled DE in Table 150 on page 1025. Proceed to “Index Files” on page 1028.

Index-Value Compression

➤ To estimate database size when the compression option is Index-value, use the formula:

```
Number of blocks * (72 bytes
+ (1.5 * database density * expanded data block size)
```

Write the result in cell labeled DE in Table 150 on page 1025. Proceed to “Index Files” on page 1028.

RLE Compression

➤ To estimate database size when the compression option is RLE, use the formula for calculating Bitmap Compression.

When the compression method is RLE, Essbase automatically uses the bitmap or Index-value method for a block if it determines that better compression can be gained. Estimating using the bitmap calculation estimates the maximum size.

Write the result in cell labeled DE in Table 150 on page 1025. Proceed to “Index Files” on page 1028.

zlib Compression

➤ To estimate database size when the compression option is zlib, use the formula for calculating Bitmap Compression.

Determining the size of a data block when zlib compression is used is difficult. Individual blocks could be larger or smaller than if compressed using other compression types. Calculating using the bitmap compression formula at least provides an approximation to use for this exercise.

Write the result in cell labeled DE in Table 150 on page 1025. Proceed to “Index Files” on page 1028.

Index Files

The calculation for the space required to store the index files (`ess`*`xxxxx`*`.ind`) uses the following factors:

- Number of existing blocks (value DB from Table 147 on page 1021)
- 112 bytes—the size of an index entry

The minimum size for the index is 8,216,576 bytes (8 MB). To calculate the size of a database index, including all index files, perform the following calculation:

```
number of existing blocks * 112 bytes = the size of database index
```

In the cell labeled DF in Table 150 on page 1025, write whichever is the larger number, the results of the calculation or 8,216,576.

Example

Assume a database with 15,000,000 blocks.

```
15,000,000 blocks
* 112
= 1,680,000,000 bytes
```

Note:

If the database is already loaded, select the Storage tab on the Database Properties window to see the size.

Fragmentation Allowance

If you are using bitmap or RLE compression, some fragmentation occurs, the amount of which is based on individual database and operating system configurations and cannot be precisely predicted.

As a rough estimate, calculate 20% of the compressed database size (value DE from Table 150 on page 1025) and write the result to the cell labeled DG in the same table.

Example

Calculating fragmentation allowance assuming a compressed database size of 5,769,000,000 bytes:

```
5,769,000,000 bytes
* .2
= 1,153,800,000 bytes
```

Outline

The space required by an outline can have two components.

- The main area of the outline is a component of both disk and memory space requirements and is calculated using the following factors:
    - The number of members in the outline
    - The length, in characters, of member names and aliases
- The attribute association area of an outline is a component only of disk space and is calculated using the following factors:
    - The number of members in each base dimension
    - The number of members in each attribute dimension

➤ To estimate the size of the outline file:

1 **Estimate the main area of the outline by multiplying the number of members by a name-length factor between 350 and 450 bytes.**

If the database includes few aliases or very short aliases and short member names, use a smaller number within this range. If you know that the member names or aliases are very long, use a larger number within this range.

Because the name-length factor is an estimated average, the following formula provides only a rough estimate of the main area of the outline.

```
number of members * name-length factor = size of main area of outline
```

Note:

See Appendix A, “Limits,” for the maximum sizes for member names and aliases.

For memory space requirements calculated later in this chapter, use the size of the main area of the outline.

2 **For disk space requirements, if the outline includes attribute dimensions, calculate the size of the attribute association area for the database. Calculate the size of this area for each base dimension. Multiply the number of members of the base dimension by the sum of the count of members of all attribute dimensions associated with the base dimension, and then divide by 8.**

Note:

Within the count of members, do not include Label Only members and shared members.

```
(number of base-dimension members * sum of count of attribute-dimension
members)/8 = size of attribute association area for a base dimension
```

3 **Sum the attribute association areas of each dimension to determine the total attribute association area for the outline.**

4 **For the total disk space required for the outline, add together the main outline area and the attribute association area, and write the result of this calculation to the cell labeled DH in Table 150 on page 1025.**

```
main area of outline + total attribute association area
```

Example

Assume the outline has the following characteristics:

- 26,000 members
- A name-length factor of 400 bytes
- A base dimension Product (23,000 members—excluding Label Only members and shared members) with two attribute dimensions associated with it—Ounces (20 members) and Pkg Type (50 members)
- A base dimension Market (2,500 members—excluding Label Only members and shared members) with one associated attribute dimension, Population (12 members)

Perform the following calculations:

1. Calculate the main area of the outline:

   ```
   name-length factor of 400 bytes * 26,000 members = 10,400,000 bytes
   ```

2. Calculate the attribute association areas:

   - For the base dimension Product:

     ```
     (23,000 * (20 + 50)) bits / 8 bits per byte = 201,250 bytes
     ```

   - For the base dimension Market:

     ```
     (2,500 * 12) bits / 8 bits per byte = 3,750 bytes
     ```

3. Sum these areas for the total attribute association area for the database:

   ```
   201,250 bytes + 3,750 bytes = 205,000 bytes
   ```

4. For a total estimate of outline disk space, add the main area of the outline and the total attribute association area:

   ```
   10,400,000 bytes + 205,000 bytes = 10,605,000 bytes (outline disk
   space requirement)
   ```

   Note:

   Do not use this procedure to calculate outline memory space requirements. Use the process described in "Outline Size Used in Memory" on page 1036.

Work Areas

Three processes create temporary work areas on the disk:

- For recovery purposes, Essbase maintains a data recovery area on the disk. The size of this area increases until the database is restructured.
- During restructuring, Essbase uses a restructuring work area on the disk.
- During upgrade from prior releases of Essbase, for recovery purposes, Essbase creates a copy of the database in a migration work area.

To create these temporary work areas, Essbase may require disk space equal to the size of the entire database. Restructuring and migration need additional work space the size of the outline. Because none of these activities occur at the same time, a single allocation can represent all three requirements.

To calculate the size of a work area used for restructuring, migration, and recovery, calculate the sum of the sizes of the following database components from Table 150 on page 1025:

- Compressed data files (value DE)
- Index files (value DF)
- Fragmentation allowance (value DG)
- Outline (value DH)

Use the following formula to calculate the size of the work area:

```
work area = size of compressed data files
+ size of index files
+ fragmentation allowance
+ outline size
```

Write the result of this calculation to the cell labeled DI in Table 150 on page 1025.

Linked Reporting Objects Considerations

You can use the Linked Reporting Objects (LROs) feature to associate objects (also called artifacts) with data cells. The objects can be flat files, HTML files, graphics files, and cell notes. See Chapter 11, "Linking Objects to Essbase Data."

Two aspects of LROs affect disk space:

- The size of the object. Because Essbase copies files used as LROs to the server, you must know the combined size of all files you are using as LROs.

  Note:

  You can set a limit on the size of a linked object, if the linked object is a file (as opposed to a cell note). See "Limiting LRO File Sizes" on page 904.

- The size of the LRO catalog, where the Essbase Kernel stores information about LROs. Cell notes and URLs are also stored in the catalog. A catalog entry is stored as an index page. For every catalog entry, Essbase uses 8 KB.

➤ To estimate the disk space requirements for LROs:

1 **Estimate the size of the objects. If a limit is set, multiply the number of LROs by that limit. Otherwise, sum the size of all anticipated LROs.**

2 **Size the LRO catalog. Multiply the total number of LROs by 8,192 bytes.**

3 **Add together the two areas and write the result of this calculation to the cell labeled DJ in Table 150 on page 1025.**

Example

Assume the database uses 1,500 LROs, which comprise the following:

- 1,000 URLs, at a maximum of 512 bytes each
- 500 cell notes

Perform the following calculations:

1. Multiply 1,000 * 512 bytes for 512,000 bytes maximum required for the stored URLs.
2. Calculate the size of the LRO catalog. Multiply 1,500 total LROs * 8,192 bytes = 12,288,000 bytes.
3. Add together the two areas; for example:

```
512,000 bytes
+ 12,288,000 bytes
= 12,800,000 bytes total LRO disk space requirement
```

Estimating the Total Essbase Server Disk Space Requirement

The earlier calculations in this chapter estimate the data storage requirement for a single database. Often, more than one database resides on the server.

In addition to the data storage required for each database, the total Essbase data storage requirement on a server includes Essbase software. Allow approximately 200 MB (209,715,200 bytes) for the base installation of Essbase software and sample applications. The allowance varies by platform and file system. See the *Oracle Hyperion Enterprise Performance Management System Installation Start Here*.

➤ To estimate the total server disk space requirement:

1. In the worksheet in Table 151 on page 1033, list the names and disk space requirements that you calculated for each database.
2. Sum the database requirements and write the total in bytes in the cell labeled DL.
3. In the cell labeled DM, write the disk space requirement for the software installed on the server; for example, 209,715,200 bytes.
4. For the total server disk space requirement in bytes, sum the values in cells DL and DM. Write this value in the cell labeled DN.
5. Convert the total in cell DN to MB by dividing the value by 1,048,576 bytes. Write this value in the cell labeled DO.

Table 151 Worksheet for Total Server Disk Space Requirement

List of Databases (From Table 150 on page 1025)	Size
a.	
b.	
c.	
d.	
e.	
f.	
g.	
Sum of database disk sizes a + b + c + d + e + f + g	DL:
Size in bytes for Essbase server software	DM:
Total Essbase Server disk requirement in bytes: DL + DM	DN:
Total Essbase Server disk requirement in megabytes (MB): DN divided by 1,048,576 bytes	DO:

Estimating Memory Requirements

The minimum memory requirement for running Essbase is 64 MB. On UNIX systems, the minimum requirement is 128 MB. Based on the number of applications and databases and the database operations on the server, the memory you require may be greater.

To estimate the memory required on Essbase Server, use the Worksheet for Total Memory Requirements, Table 155 on page 1044, to collect and total server memory requirements. To calculate the requirements for this worksheet, review the following topics.

Estimating Startup Memory Requirement for Applications

You must calculate overall memory used at application startup plus the memory requirements for each database.

Each open application has the following memory requirements at startup:

- Essbase code and static data (10 MB). This number may be more or less, depending on the operating system.
- (**Optional**) JVM (2 to 4 MB), which is used for custom-defined functions. This value depends on the operating system.

Multiply the number of applications that will be running simultaneously on the server by the startup requirement and write the resulting value in the cell labeled ML in Table 155 on page 1044.

Estimating Startup Memory Requirements for Databases

Calculate memory requirements for each database on Essbase Server.

For each database, make a copy of Table 155 or use a separate sheet of paper as a worksheet for one database. If multiple databases are on Essbase Server, repeat this process for each database. Write the database name on the worksheet.

Each row links to information that describes how to size that component. Perform each calculation and note the results in the appropriate cell in the Size column. Some calculations use the factors that you wrote in Table 153 on page 1035. After filling in all the sizes in the Size column, total them to determine the memory requirement for that database.

After estimating disk space for all databases on the server, proceed to "Estimating the Total Essbase Server Disk Space Requirement" on page 1032.

Table 152 Worksheet for Estimating Memory Requirements for a Database

Database Name	**Size (MB)**
Memory Requirement:	
Startup requirements per database:	
Database outline See "Outline Size Used in Memory" on page 1036.	MA:
Index cache See "Sizing Caches" on page 826.	MB:
Cache-related overhead See "Cache-Related Overhead" on page 1037.	MC:
Area for data structures See "Memory Area for Data Structures" on page 1038.	MD:

Database Name	Size (MB)
Operational Requirements:	
Memory used for data retrievals See "Estimating Additional Memory Requirements for Data Retrievals" on page 1039.	ME:
Memory used for calculations See "Estimating Additional Memory Requirements for Calculations" on page 1043.)	MF:
Summarize the size values from MA through MF for an estimate of the total memory required for a database.	MG:
Divide the value from MG by 1,048,576 bytes for the total database memory requirement in megabytes (MB).	MH:

In Table 155 on page 1044, enter the name of the database and the total memory requirement in MB, MH.

Factors to Be Used in Sizing Memory Requirements

Before you start the estimate, calculate factors to be used in calculating the estimate.

Table 153 on page 1035 lists sizing factors with references to sections in this chapter and other chapters that provide information to determine these sizes. Go to the section indicated, perform the calculation, and return to Table 153. Write the size, in bytes, in the Value column this table.

Later in this chapter, you can refer to Table 153 for values to use in various calculations.

Table 153 Factors Used to Calculate Database Memory Requirements

Database Sizing Factor	Value
The number of cells in a logical block See "The Number of Cells in a Logical Block" on page 1037.	MI:
The number of threads allocated through the SERVERTHREADS ESSCMD See the *Oracle Essbase Technical Reference*.	MJ:
Potential stored-block size See "Size of Expanded Data Block" on page 1023.	MK:

The calculations in this chapter do not account for other factors that affect how much memory is used. The following factors have complex implications and are not included in the discussion:

- Cache memory locking. Whether ache memory locking is enabled affects how the operating system and Essbase manage memory. See "Deciding Whether to Use Cache Memory Locking" on page 826.
- Different operation types and their associated cache allocations. Data load, data retrieval, and calculation operations set aside memory for the data file, data, and calculation caches, plus some overhead associated with the caches.

- The sizes of the retrieval buffer and the retrieval sort buffer. See "Changing Buffer Size" on page 899 for a discussion of the significance of the size of the retrieval buffer and the retrieval sort buffer.
- Database workload; for example, the complexity of a calculation script or the number and complexity of data queries.
- The number of data blocks defined using the CALCLOCKBLOCK setting in the `essbase.cfg` file in combination with the SET LOCKBLOCK setting, which specifies which CALCLOCKBLOCK setting to use.
- The number of Dynamic Calc members in the outline, including members of attribute dimensions.
- The isolation level settings
- Synchronization points
- Use of triggers
- MDX implications

Outline Size Used in Memory

The attribute association area included in disk space calculations is not a sizing factor for memory. Calculate only the main area of the outline.

For memory size requirements, outline size is calculated using the following factors:

- The number of members in the outline
- The length, in characters, of member names and aliases

➤ To calculate the outline memory requirement, multiply the number of members by a name-length factor of 300 bytes–400 bytes and write the result in the cell labeled MA in Table 152 on page 1034.

If the database includes few aliases or very short aliases and short member names, use a smaller number in the 300 byte–400 byte range. If you know that the names or aliases are very long, use a larger number within this range.

Because the name-length factor is an estimated average, the following formula provides only a rough estimate of the main area of the outline:

```
memory size of outline
= number of members
* name-length factor
```

Note:

See Appendix A, "Limits," for the maximum sizes for member names and aliases.

Example

Assuming that the outline has 26,000 members and a median name-length, use the following calculation to estimate the outline size used in memory:

```
26,000 members
* 350 bytes per member
= 9,100,000 bytes
```

Index Cache

At startup, Essbase sets aside memory for the index cache, the size of which can be user-specified. To determine the size of the index cache, see "Sizing Caches" on page 826 and write the size in the cell labeled MB in Table 152 on page 1034.

Cache-Related Overhead

Essbase uses additional memory while it works with the caches.

The calculation for this cache-related overhead uses the following factors:

- Index cache (value MB from Table 152 on page 1034)
- The number of server threads (value MJ from Table 153 on page 1035)

➤ To calculate the cache-related overhead at startup:

1. **Calculate half the index cache size, in bytes.**

```
index cache size
* .5
= index cache-related overhead
```

2. **Calculate additional cache overhead in bytes using the following formula:**

```
((# of server threads allocated to the Essbase Server process * 3)
* 256)
+ 5242880 bytes
= additional cache overhead
```

3. **Sum the index cache overhead plus the additional cache overhead. Write the result to the cell labeled MC in Table 152 on page 1034.**

```
cache-related overhead
= index cache-related overhead
+ additional cache overhead
```

The Number of Cells in a Logical Block

The term logical block refers to an expanded block in memory.

➤ To determine the cell count of a logical block, multiply all members of each dense dimension (including Dynamic Calc and Dynamic Calc and Store members but excluding Label Only and shared members):

1. Using Table 154 on page 1038 as a worksheet, enter each dense dimension and its number of members, excluding Label Only and shared members. If there are more than seven dense dimensions, list the dimensions elsewhere and include all dense dimensions in the calculation.

2. Multiply the number of members of the first dense dimension (line a.) by the number of members of the second dense dimension (line b.) by the number of members of the third dense dimension (line c.), and so on, to determine the total number of cells in a logical block. Write the result to the cell labeled MI in Table 153 on page 1035.

   ```
   a * b * c * d * e * f * g = total number of cells
   ```

Table 154 Determining the Number of Cells in a Logical Block

Enter Dense Dimension Name	Number of Members
	a.
	b.
	c.
	d.
	e.
	f.
	g.

Example

Excluding Label Only and shared members, the dense dimensions in Sample.Basic contain 17 (Year), 14 (Measures), and 4 (Scenario) members. The calculation for the cell count of a logical block in Sample.Basic:

```
17 * 14 * 4 = 952 cells
```

Memory Area for Data Structures

At application startup, Essbase sets aside an area of memory based on the following factors:

- The number of members in the outline
- The number of cells in a logical block (value MI in Table 153 on page 1035)
- The number of threads on the server (value MJ in Table 153)

➤ To calculate the data structure area in memory:

1. Use the following formula to calculate the size in bytes:

```
Number of threads
* ((Number of members in the outline * 26 bytes)
+ (Logical block cell count * 36 bytes))
```

2 **Write the result in the cell labeled MD in Table 152 on page 1034.**

Example

Assuming 20 threads for the Sample.Basic database, the startup area in memory required for data structures is calculated:

```
20 threads
* ((79 members * 26 bytes) + (952 cells * 36 bytes))
= 726,520 bytes 726,520 bytes
/ 1,048,576 bytes = .7 MB
```

Estimating Additional Memory Requirements for Database Operations

In addition to startup memory requirements, operations such as queries and calculations require additional memory. Because of many variables, the only way to estimate memory requirements of operations is to run sample operations and monitor the memory used during them. The following sections provide guidelines.

Estimating Additional Memory Requirements for Data Retrievals

Essbase processes requests for database information (queries) from many sources. For example, Essbase processes queries from Oracle's Hyperion® Essbase® Spreadsheet Toolkit and from Report Writer. Essbase uses additional memory when it retrieves the data for these queries, especially when Essbase must perform dynamic calculations to retrieve the data. This section describes Essbase memory requirements for query processing.

Essbase is a multi-threaded application in which queries get assigned to threads. Threads are automatically created when Essbase is started. In general, a thread exists until you shut down Essbase Server. See "Multithreading" on page 688.

As Essbase processes queries, it cycles through available threads. For example, assume that 20 threads are available at startup. As each query is processed, Essbase assigns each succeeding query to the next sequential thread. After it has assigned the 20th thread, Essbase cycles back to the beginning, assigning the 21st query to the first thread.

While processing a query, a thread allocates some memory, and then releases most of it when the query is completed. Some memory is released to the operating system, and some is released to the dynamic calculator cache for the database being used. However, the thread holds on to a portion of the memory for possible use in processing subsequent queries. As a result, after a thread has processed its first query, the memory held by the thread is greater than it was when Essbase first started.

Essbase uses the maximum memory for query processing when both conditions are true:

- The maximum simultaneous queries that will occur are being processed.

- All threads have been assigned to at least one query by Essbase.

In the example where 20 threads are available at startup, the maximum amount of memory is used for queries when at least 20 queries have been processed and the maximum number of simultaneous queries are in process.

Calculating the Maximum Additional Memory Required

➤ To estimate query memory requirements by observing actual queries:

1. **Observe memory use during queries.**
2. **Calculate the maximum possible use of memory for query processing by totaling the memory used by queries that will be run simultaneously, and add the memory acquired by threads that are now waiting for queries.**

   Use the following variables when you calculate the formula in Estimating the Maximum Memory Usage for a Query Before and After Processing:

   - Total number of threads (`Total#Threads`)
   - Maximum memory usage for any query *during* processing (`MAXAdditionalMemDuringP`)
   - Maximum number of possible concurrent queries (`Max#ConcQueries`)
   - Maximum memory usage for any query *after* processing (`MAXAdditionalMemAfterP`)

Determining the Total Number of Threads

The potential number of threads available is based on the number of licensed ports that are purchased. The actual number of threads available depends on settings that you define for the Agent or the server. Use the number of threads on the system as the value for `Total#Threads` in later calculations.

Estimating the Maximum Number of Concurrent Queries

Determine the maximum number of concurrent queries and use this value for `Max#ConcQueries` in later calculations. This value cannot exceed the value for `Total#Threads`.

Estimating the Maximum Memory Usage for a Query Before and After Processing

The memory usage of individual queries depends on the size of each query and the number of data blocks that Essbase needs to access to process each query. To estimate the memory usage, calculate the additional memory Essbase uses during processing and after processing each query.

Choose several queries that you expect to use the most memory. Consider queries that must process large numbers of members; for example, queries that perform range or rank processing.

➤ To estimate the memory usage of a query:

1. **Turn the dynamic calculator cache off by setting the `essbase.cfg` setting DYNCALCCACHEMAXSIZE to 0 (zero). Turning off the dynamic calculator cache enables measurement of memory still held by a thread by**

ensuring that, after the query is complete, the memory used for blocks during dynamic calculations is released by the `ESSSVR` process to the operating system.

2 **Start the Essbase application.**

3 **Using memory monitoring tools for the operating system, note the memory used by Essbase Server before processing the query. Use the value associated with the `ESSSVR` process.**

   Use this value for *`MemBeforeP`*.

4 **Run the query.**

5 **Using memory monitoring tools for the operating system, note the peak memory usage of Essbase Server while the query is processed. This value is associated with the `ESSSVR` process.**

   Use this value for *`MemDuringP`*.

6 **Using memory monitoring tools for the operating system, after the query is completed, note the memory usage of Essbase. This value is associated with the `ESSSVR` process.**

   Use this value for *`MemAfterP`*.

7 **Calculate the following two values:**

   - Additional memory used while Essbase processes the query (*`AdditionalMemDuringP`*):

     `AdditionalMemDuringP = MemDuringP - MemBeforeP`

   - Additional memory used after Essbase has finished processing the query (*`AdditionalMemAfterP`*):

     `AdditionalMemAfterP = MemAfterP - MemBeforeP`

8 **When you have completed the above calculations for all the relevant queries, compare all results to determine the following two values:**

   - The maximum *`AdditionalMemDuringP`* used by a query: (*`MAXAdditionalMemDuringP`*)
   - The maximum *`AdditionalMemAfterP`* used by a query: (*`MAXAdditionalMemAfterP`*)

9 **Insert the two values from step 7 into the formula in the following statement.**

   The additional memory required for data retrievals will not exceed the following:

```
Max#ConcQueries
* MAXAdditionalMemDuringP
+ (Total#Threads - Max#ConcQueries)
* MAXAdditionalMemAfterP
```

   Write the result of this calculation, in bytes, to the cell labeled ME in Table 152 on page 1034.

   Because this calculation method assumes that all of the concurrent queries are maximum-sized queries, the result may exceed your actual requirement. It is difficult to estimate the actual types of queries that will be run concurrently.

   To adjust the memory used during queries, you can set values for the retrieval buffer and the retrieval sort buffer. See "Setting the Retrieval Buffer Size" on page 899 and "Setting the Retrieval Sort Buffer Size" on page 900.

Estimating Additional Memory Requirements Without Monitoring Actual Queries

If you cannot perform this test with actual queries, you can calculate a rough estimate for operational query requirements. Requirements for each retrieval vary considerably. Generally, this estimate uses the following fixed factors:

- The size of the retrieval buffer (10,240 bytes)
- If sorting is involved in the query, the size of the retrieval sort buffer (20,480 bytes)
- The size of the buffer that holds formatting information (140 KB, which rounds up to 144,000 bytes)
- Report Writer factors:
  - 40 bytes per selected member in the retrieval
  - 8 bytes per member of the largest dimension

This estimate also uses the following variables:

- The memory used for dynamically calculated values, which is based on the following numbers:
  - The number of blocks specified by the SET LOCKBLOCK command.
  - The number of cells in a logical block, which includes Dynamic Calc members. See value MH in Table 153 on page 1035.
- The size of the data cache. See "Sizing the Data Cache" on page 829.
- If direct I/O is used, the size of the data file cache. See "Sizing the Data File Cache" on page 828.
- Report Writer factors the number of:
  - Selected members in an average retrieval.
  - Members in the largest dimension to calculate

You can then use the following two calculations for the memory needed for retrievals:

- Buffer and work area used in each retrieval:

  ```
  retrieval buffer (10,240 bytes)
  + retrieval sort buffer (20,480 bytes)
  + formatting buffer (144,000 bytes)
  + each selected member in the retrieval (40 bytes)
  + each member of the largest dimension (8 bytes)
  + dynamic calc area
  + data cache size
  + data file cache size
  ```

- Memory needed for estimated number of concurrent retrievals:

  ```
  Member storage area for the largest dimension
  + (number of retrievals
  * sum of buffer and work areas used in each retrieval)
  ```

Summarize the calculations and write the result, in bytes, to the cell labeled ME in Table 152 on page 1034.

Example

To estimate the maximum memory needed for concurrent queries, assume the following values for this example:

- The area required by the largest dimension:

  ```
  23,000 members * 8 bytes/member = 184,000 bytes
  ```
- The buffer and work area values apply for each retrieval:
  - Retrieval buffer: 10,240 bytes
  - Retrieval sort buffer: 20,480 bytes
  - Formatting buffer: 144,000 bytes
  - Dynamic calc area: 761,600 bytes

    With SET LOCKBLOCK set as 100 blocks, the calculation:

    ```
    100 blocks * 7616-byte block size = 761,600 bytes
    ```
  - Data cache size: 3072 KB, which equals 3,145,728 bytes
  - Data file cache size: zero. Direct I/O is not used in the example.
- The largest dimension has 23,000 members.
- The maximum number of concurrent queries is 20.
- The number of selected members is generalized across all queries to be 10,000 members. The approximate memory requirement equals the following:

  ```
  10000 members * 40 bytes/member = 400,000 bytes
  ```

Estimated memory for retrievals:

```
184,000 bytes + (20 concurrent inquiries
* (10,240 bytes + 20,480 bytes + 144,000 bytes
+ 761,600 bytes + 3,145,728 bytes + 400,000 bytes))
= 75,824,960 bytes
```

Estimating Additional Memory Requirements for Calculations

For existing calculation scripts, you can use the memory monitoring tools provided for the operating system on the server to observe memory usage. Run the most complex calculation and notice the memory use before and while running the calculation. Calculate the difference and use that figure as the additional memory requirement for the calculation script.

For a comprehensive discussion of calculation performance, see Chapter 55, "Optimizing Calculations."

If you cannot perform a test with a calculation script, you can calculate a very rough estimate for the operational requirement of a calculation by adding the following values:

- The size of the calculator cache.

  See "Sizing the Calculator Cache" on page 830.
- The size of the data cache.

  See "Sizing the Data Cache" on page 829.

- The size of the outline. For calculations, Essbase uses approximately 30 additional bytes of memory per member of the database outline. For information about concurrent calculations, see "Managing Caches to Improve Performance" on page 881.
- The size of the memory area used by the blocks set aside by the SET LOCKBLOCK command.
    - To calculate the memory requirement in bytes, multiply the specified number of data blocks by the logical size of the data blocks.
    - For the logical block size, multiply the number of cells (value MI in Table 153 on page 1035) by 8 bytes per cell.

For the total calculation requirement, summarize the memory needed for all calculations that will be run simultaneously and write that total to the cell labeled MF in Table 152 on page 1034.

Note:

The size and complexity of the calculation scripts affect the memory required. The effects are difficult to estimate.

Estimating Total Essbase Memory Requirements

You can use Table 155 as a worksheet on which to calculate an estimate of the total memory required on the server.

Table 155 Worksheet for Total Server Memory Requirement

Component	Memory Required, in Megabytes (MB)
Sum of application startup memory requirements See "Estimating Startup Memory Requirement for Applications" on page 1034.	ML:
In rows a–g, list concurrent databases (from copies of Table 152 on page 1034) and enter their respective memory requirements (MH) in the column to the right.	
a.	MH:
b.	MH:
c.	MH:
d.	MH:
e.	MH:
f.	MH:
g.	MH:
Operating system memory requirement	MN:
Total estimated memory requirement for the server	MO:

➤ To estimate the total Essbase memory requirement on a server:

1. In the cell labeled ML, record the total startup memory requirement for applications, as described in "Estimating Startup Memory Requirement for Applications" on page 1034.
2. List the largest set of databases that will run concurrently on the server. In the Memory Required column, for the MH value for each database, note the memory requirement estimated in the database requirements worksheet, Table 152 on page 1034.
3. Determine the operating system memory requirement, and write the value in megabytes to the cell labeled MN in Table 155.
4. Total all values and write the result in the cell labeled MO.
5. Compare the value in MN with the total available random-access memory (RAM) on the server.

Note:

In addition, consider memory requirements for client software, such as Oracle Essbase Administration Services, that may be installed on the Essbase Server computer. See the *Oracle Hyperion Enterprise Performance Management System Installation Start Here*.

If cache memory locking is enabled, the total memory requirement should not exceed two-thirds of available RAM; otherwise, system performance can be severely degraded. If cache memory locking is disabled, the total memory requirement should not exceed available RAM.

If insufficient memory is available, you can redefine the cache settings and recalculate the memory requirements. This process can be iterative. See "Fine-Tuning Cache Settings" on page 838. In some cases, you may need to purchase additional RAM.

E Using ESSCMD

In This Appendix

Also see:

- *Oracle Essbase Technical Reference* for ESSCMD syntax and usage
- *Oracle Essbase Administration Services Online Help* for information about MaxL Script Editor

Understanding ESSCMD

With ESSCMD, you can execute Essbase Server operations at the command line, in either interactive or batch mode.

Interactive mode means entering commands at the ESSCMD command line and receiving prompts where necessary. Interactive mode is convenient for short operations that require few commands, checking for information on the fly, and error checking.

Batch processing mode is used for automating routine Essbase Server maintenance and diagnostic tasks. You can write a script or batch file and run it from the command line. Batch processing mode is convenient if you frequently use a particular series of commands, or if a task requires many commands.

Understanding Syntax Guidelines

In general, use the same syntax for entering ESSCMD commands that you do for other calculation commands. However, differences exist between the ESSCMD interactive and batch processing modes in the requirements for quotation marks and the semicolon statement terminator. Use the guidelines in this section when creating script or batch files.

Quotation Marks

Quotation marks (" ") enclose character parameters and responses to commands.

- In interactive ESSCMD, using quotes is optional. Use quotes when a parameter has an embedded space; for example:

  ```
  CALC "Calc All;"
  ```

- In an ESSCMD script file, always enclose all character parameters and responses to commands in quotation marks; for example:

  ```
  LOGIN "Localhost" "user1" "Password";
  ```

- Numeric parameters and responses do not require quotation marks.
- Do not enclose quotation marks within quotation marks.

Semicolon Statement Terminator

The ; (semicolon) statement terminator signals the end of a command; for example:

```
SELECT "SAMPLE" "BASIC";
```

- In interactive ESSCMD, pressing the Enter key signals ESSCMD that the command is complete. The statement terminator is optional.
- In an ESSCMD script file, you should use the terminator, even though it is optional, if a command has many parameters. Doing so is especially important to signal the end of the parameter list if some parameters are optional.

If you omit some optional parameters and do not use a semicolon to end the list, ESSCMD looks for the remaining values in the next command in the file, leading to unpredictable results.

The SETAPPSTATE and SETDBSTATE commands, defined in the *Oracle Essbase Technical Reference*, are examples of commands that you should terminate with a semicolon to prevent confusion in processing.

Note:

All syntax examples in this chapter use quotation marks and semicolon terminators.

Canceling ESSCMD Operations

When running ESSCMD, you can cancel an asynchronous operation, such as a calculation, export, or restructure operation, by pressing and holding the Esc key until ESSCMD responds.

Referencing Files

Some commands require that you precede artifact or file names with a numeric parameter, from 1 to 4, that tells Essbase where to look for the artifact or file. The parameter directs ESSCMD to look for files in other applications, databases, or systems.

Table 156 lists each value for the numeric parameter (*Number*), the file location to which it applies, and the information that ESSCMD requests when you use each parameter setting. *appname* is the application name and *dbname* is the database name.

Table 156 List of Numeric Parameters

Number	File	Essbase prompts user for:
1	Local or client-based file	• Windows: Files in the *ARBORPATH*/client/*appname*/*dbname* directory • UNIX: Files in the *ARBORPATH*/client/*appname*/*dbname* directory
2	Remote or server-based file	• Windows: Files in the *ARBORPATH*/app/*appname*/*dbname* directory • UNIX: Files in the *ARBORPATH*/app/*appname*/*dbname* directory
3	File	Fully qualified path to the file, unless file is in the current ESSCMD directory
4	SQL table	Full network and database information for the SQL table

For example, the LOADDATA command can load a data file that resides on the client or Essbase Server. The command requires the numeric parameter to tell Essbase where to look for the data file. This example causes ESSCMD to prompt for the fully qualified path name of the file to load:

```
LOADDATA 3
```

File extensions are usually optional in interactive and batch processing modes, except when using commands that require a numeric parameter that indicates the location of files:

- If you use file option 3 (File), you must enter the file extension in both interactive and batch processing modes.
- If the artifact is in the directory from which you started ESSCMD, you need not enter a path.

Accessing Multiple Databases

Because ESSCMD supports multiple login instances on Essbase Server, you can access more than one database in a single session. Even when you log in to multiple databases, you use only one port on your server license.

Considering Case-Sensitivity

The Essbase Server creates application and database names exactly as the user specifies. Case is not changed for any platform.

For backward compatibility, the Essbase Server searches for existing application and database names using the exact case first. However, if it cannot find the file, Essbase Server searches all possible case combinations to find the existing application and database names.

Essbase does not allow you to create application and database names that differ only in case. For example, Essbase displays an error message if application mYdATA exists and you try to create an application MyData.

You can choose to make member names case-sensitive.

➤ To make a member case-sensitive, see “Making Members Case-Sensitive” in the *Oracle Essbase Administration Services Online Help*.

Getting Help

For descriptions and syntax for individual ESSCMD commands, see the *Oracle Essbase Technical Reference*.

Preparing to Start ESSCMD

Before you start ESSCMD, ensure that the following items are installed and running:

- Essbase
- Communications protocol (TCP/IP) on Essbase

Starting and Quitting ESSCMD

ESSCMD comprises two files: `esscmd.exe` and `esscmd.hlp` (`esscmd` and `esscmd.hlp` on UNIX platforms) on Essbase Server and Essbase Administration Server.

➤ To start ESSCMD, enter ESSCMD at the operating system command prompt.

When ESSCMD starts, a command prompt is displayed:

```
:::[n]->
```

where *n* is the value of the active login instance. Each subsequent, successful login increments this value by one. When ESSCMD starts, the instance number is zero (0).

Note:

Use the SETLOGIN command to toggle between active login instances. Use the LISTLOGINS command to view active login instances.

➤ To quit ESSCMD, enter EXIT at the prompt and press Enter.

ESSCMD disconnects from the application server and terminates the session.

Using Interactive Mode

In interactive mode, you enter commands and respond to prompts, which is useful when you are performing simple tasks that require few commands. If you are performing more complex tasks that require many commands, consider creating a script file or batch file. See "Using Script and Batch Files for Batch Processing" on page 1052.

For syntax conventions when working in interactive mode, see "Understanding Syntax Guidelines" on page 1047.

Logging on to Essbase Server

After starting ESSCMD, you must connect to Essbase Server so that you can enter commands.

➤ To log on to Essbase Server:

1 **At the ESSCMD prompt, log in to the server with the LOGIN command.**

See the *Oracle Essbase Technical Reference*.

2 **Enter the server name.**

When you connect from the server console, the server name depends on your network setup. For example, the name could be `aspen`.

3 **Enter your user name.**

4 **Enter your password.**

The ESSCMD prompt is displayed:

```
aspen:::userName[1]->
```

`userName` is your login name.

You can enter any valid ESSCMD command. For a complete listing of commands, see the*Oracle Essbase Technical Reference*.

Note:

To load an application into memory and select a database on the Essbase server, use the SELECT command to select a database from an application that resides on the server.

The ESSCMD prompt is displayed:

```
aspen:appname:dbname:userName[1]->
```

`appname` is the name of the application. *`dbname`* is the name of the database to which you are connected.

Entering Commands

Choose either of the following methods to enter commands in interactive mode:

- Enter the command and press Enter.

  ESSCMD prompts you for each of the command parameters. For example, the SELECT command has two parameters, as shown in the command syntax:

```
SELECT " appname " " dbname ";
```

If you enter only SELECT and press Enter, EESSCMD prompts you for the first parameter, the application name (*appname*). After you enter the application name and press Enter, ESSCMD prompts you for the database name (*dbname*).

- Type the commands and all parameters, and press Enter.

  Using SELECT as the example, you would enter:

  ```
  SELECT "Sample" "Basic";
  ```

Whichever method you use, the interactive prompt now reflects the application and database names. For example, the following prompt tells you that the Sample application and Basic database are selected:

```
aspen:Sample:Basic:User[1]->
```

In this case, you can enter other commands without the application or database name parameters that it normally requires.

Canceling Operations

While ESSCMD is running, you can cancel an asynchronous operation, such as a calculation, export, or restructure operation, by pressing and holding the Esc key until ESSCMD responds.

Caution!

Do not pause or suspend your system while Essbase is processing a command. Pausing the system may prevent Essbase from correctly completing the command.

Using Script and Batch Files for Batch Processing

If you use a series of commands frequently or you must enter many commands to complete a task, consider automating the task with a script or batch file. These files are useful for batch data loads and complex calculations.

See "Understanding Syntax Guidelines" on page 1047.

- A *script file* contains ESSCMD commands. You can run a script file from the operating system command line or from within an operating system batch file, and the script file is processed by ESSCMD. By default, an ESSCMD script file has a `*.SCR` file extension. You can use a different extension.
- A *batch file* is an operating system file that calls multiple ESSCMD scripts and can also include operating system commands. You can use a batch file to run multiple sessions of ESSCMD. You can run a batch file on Essbase Server from the operating system prompt; the file is processed by the operating system. On Windows, batch files have `*.bat` file extensions.

Note:

On UNIX, a batch or script file is written as a shell script. A shell script usually has the file extension `.sh` (Bourne or Korn shell) or `.csh` (C shell).

When you run a script or batch file, ESSCMD executes the commands in sequence until it reaches the end of the file.

Some commands may be changed in new releases of Essbase. Changes might affect existing scripts. To ensure that your scripts work correctly in the current release, see the *Oracle Essbase New Features* booklet and the Essbase Readme for information about changed or deleted commands and change your scripts if needed.

Writing Script Files

ESSCMD script files automate an often-used or lengthy series of commands. Each script file must be a complete ESSCMD session, with login, application and database selection, logout, and termination commands.

➤ To define a script file:

1. **Enter ESSCMD commands in any editor that saves data in ASCII text.**
2. **Save the file with the `.scr` ESSCMD script file extension.**

   For example, the following script file, test`.scr`, was created in Notepad:

```
LOGIN "localhost" "User1" "password";
SELECT "Sample" "Basic";
GETDBSTATE
EXIT;
```

   When run from the operating system command line, this script logs User1 into the Essbase localhost server, selects the Sample application and Basic database, gets database statistics, and quits the ESSCMD session.

Running Script Files

➤ To run script files in ESSCMD:

1. **Enter the following command at the operating system prompt:**

```
ESSCMD scriptFileName.scr
```

2. **Replace *`scriptFileName`* with the name of the script file. For example, enter the following if the script file is in the current directory:**

```
ESSCMD TEST.SCR
```

3. **If the script file is not in the current directory, include the path. For example:**

```
ESSCMD C:\WORK\SCRIPTS\TEST.SCR (an absolute path on Windows)
```

or

```
ESSCMD..\SCRIPTS\TEST.SCR (a relative path on Windows)
```

Handling Command Errors in Script Files

ESSCMD error-handling features provide error checking and handling for your script files. You can write error-handling commands into your script file to check for errors and, if necessary, branch to an appropriate error-handling response.

After each ESSCMD command is executed, a number is stored in an internal buffer. If the command executes successfully, `0` is returned to the buffer. If the command is unsuccessful, the error number is stored in the buffer; a state called *nonzero status*.

For error checking within an ESSCMD script file, ESSCMD provides the following error-handling commands:

- IFERROR checks the previously executed command for a nonzero return status (failure to execute). If the status is not zero, processing skips all subsequent commands and jumps to a user-specified point in the file, where it resumes. The script file can branch to an error-handling routine or the end of the file.
- RESETSTATUS reverts all saved status values to 0 (zero) in preparation for more status checking.
- GOTO forces unconditional branching to a user-specified point in the file, whether or not an error occurred.

In this `load.scr` example file, if the LOADDATA command does not execute successfully, EESSCMD branches to the end of the file to avoid attempting to calculate and run a report script on the empty database.

```
LOGIN "localhost" "User1" "password" "Sample" "Basic";
LOADDATA 2 "calcdat";
IFERROR "Error";
CALC "Calc All;";
IFERROR "Error";
RUNREPT 2 "Myreport";
IFERROR "Error";
[possible other commands]
EXIT;

:Error
```

Note:

You can use the OUTPUT command to log errors to a text file.

For the syntax and usage of ESSCMD error commands, see the *Oracle Essbase Technical Reference*.

Reviewing Sample Script Files

Essbase provides sample script files, based on the Sample.Basic database, that demonstrate common batch operations. The sample script files reside in the following directory:

ARBORPATH/app/Sample/Basic

Sample Script: Importing and Calculating Data

The sample1.src file executes the following actions:

- Logs on to Essbase Server
- Selects an application and database
- Prevents other users from logging on and making changes to the database
- Imports data from a text file
- Calculates the database
- Reenables logins
- Exits ESSCMD

```
LOGIN "Poplar" "TomT" "Password";
SELECT "Sample" "Basic";
DISABLELOGIN;
IMPORT 2 "ACTUALS" 4 "Y" 2 "ACTUAL" "N";
CALCDEFAULT;
ENABLELOGIN;
EXIT;
```

Sample Script: Building Dimensions and Importing and Calculating Data from a SQL Source

The sample2.scr file executes the following actions:

- Logs on to Essbase Server
- Selects an application and database
- Prevents other users from logging on and making changes to the database
- Updates the outline from an SQL data source
- Imports data from SQL
- Calculates the database
- Reenables logins
- Exits ESSCMD

```
LOGIN "Poplar" "TomT" "Password";
SELECT "Sample" "Basic";
DISABLELOGIN;
BUILDDIM 2 "PRODRUL" 4 "PRODTBL" 4 "PROD.ERR";
IMPORT 4 "TOMT" "PASSWORD" 2 "ACTUAL" "N";
CALCDEFAULT;
EXIT;
```

Sample Script: Scheduling Report Printing

The `sample3.scr` file executes the following actions:

- Logs on to Essbase Server
- Selects an application and database
- Assigns reports that output to files for later printing
- Exits ESSCMD

```
LOGIN "Poplar" "TomT" "Password";
SELECT "Sample" "Basic";
RUNREPT 2 "REP1" "REP1.OUT";
RUNREPT 2 "REP2" "REP2.OUT";
RUNREPT 2 "REP3" "REP3.OUT";
EXIT;
```

Writing Batch Files

You can write a batch file that runs one or more ESSCMD scripts and includes operating system commands. See your operating system instructions to learn the syntax for writing batch files.

Handling Command Errors in Batch Files

For the operating system batch file, you can use ESSCMD command return values to control the flow of scripts that the batch file executes.

An ESSCMD program returns an integer value upon exiting. This value represents the status of the last executed command, usually whether the command succeeded or failed. You can set up your batch file to test for this value, and if the test fails, branch to an error-handling response. This process is similar to creating a script file. See “Handling Command Errors in Script Files” on page 1054.

For example, a batch file could contain three scripts: an ESSCMD batch file that loads data, a calculation script that performs calculations on the data, and a report script that reports on the results of the calculation. If the load batch file fails, the calculations and reporting also fail. In this case, it would be best to stop the batch file after the failure of the load file and correct the error that caused the failure before continuing. If your batch file tests for the return value of the load process, and this return value indicates failure, the batch file can jump to the end of the file and stop or execute some other error-handling procedure, rather than attempting to calculate data that did not load.

The following example shows a Windows operating system batch file and the contents of one of the ESSCMD scripts it runs, `load.scr`. Because error-checking requirements vary, the syntax in this example may not correspond to that of your operating system. See your operating system documentation for error checking in batch files.

An operating system batch file could contain commands like this:

```
ESSCMD LOAD.SCR
If not %errorlevel%==0 goto Error
ESSCMD CALC.SCR
```

```
If not %errorlevel%==0 goto Error
ESSCMD REPORT.SCR
If not %errorlevel%==0 goto Error
Echo All operations completed successfully
EXIT

:Error
Echo There was a problem running the script
```

F

Naming Restrictions for Essbase Applications, Databases, and Members

In This Appendix

Naming Restrictions for Applications and Databases

When naming applications and databases, follow these rules:

- Use no more than 8 bytes when naming non-Unicode-mode applications and databases.
- Use no more than 30 characters when naming Unicode-mode applications and databases.
- Do not use spaces in the name.
- Do not use the characters listed in Table 157 in the name:

Table 157 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 not case-sensitive. However, on case-sensitive file systems, the application or database name is created exactly as you enter it. Therefore, when creating, renaming, or copying applications and databases on case-sensitive file systems, Essbase ensures that the same application or database name but with different case usage cannot be used. For example, if you create an application name with all uppercase letters (NEWAPP), you cannot then create an application with the same name but with mixed-case letters (Newapp). Also, when manually copying application and database files from one computer to another and then creating an application or database, you must use the same case for the application and database directory names on both computers.

Naming Restrictions for Dimensions, Members, and Aliases

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

- Use no more than 80 bytes when naming non-Unicode-mode dimensions, members, or aliases.
- Use no more than 80 characters when naming Unicode-mode dimensions, members, or aliases.
- 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 matching names with only case differences for a dimension name. For example, do not name two dimensions “Product” and “product.”

- Do not use quotation marks (" "), brackets ([ ]), or tabs in dimension, member, or alias names.
- At the beginning of dimension or member names, do not use the characters listed in Table 158:

Table 158 Restricted Characters for Dimension, Member, and Alias Names

Character	Description
@	at sign
\	backslash
{ }	brace
,	comma
-	dash, hyphen, or minus
=	equal sign
<	less than sign
()	parentheses
.	period
+	plus sign
'	single quotation mark
_	underscore
|	vertical bar

- Do not place spaces at the beginning or end of names. Essbase ignores such spaces.
- Do not use these words as dimension or member names:
  - Calculation script commands, operators, and keywords
  - Report writer commands
  - Function names and function arguments.
  - Names of other dimensions and members (unless the member is shared), and generation names, level names, and aliases in the database
  - Any of these words:

```
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
```

Note:

If you enable Dynamic Time Series members, do not use the associated generation names—History, Year, Season, Period, Quarter, Month, Week, or Day.

Using Dimension and Member Names in Calculation Scripts, Report Scripts, Formulas, Filters, Substitution Variable Values and Environment Variable Values

In substitution variable values, environment variable values, calculation scripts, report scripts, filter definitions, partition definitions, or formulas, you must enclose member names in brackets ([ ]) when used within MDX statements, and otherwise in quotation marks (" "), in these situations:

- Name starts with one or more numerals (for example, 100).
- Name contains spaces or any characters listed in Table 159:

Table 159 Characters that Require Member Name Enclosures

Character	Description
&	ampersand
*	asterisk
@	at sign
\	backslash
{ }	braces
:	colon
,	comma
-	dash, hyphen, or minus
!	exclamation point
=	equal sign
>	greater than sign
<	less than sign
()	parentheses
%	percent sign
.	period
+	plus sign

Character	Description
;	semicolon
/	slash
~	tilde

In calculation scripts and formulas, you must enclose these member names, which are also Oracle Essbase keywords, in quotation marks (" ") for block storage databases, and in brackets ([ ]) for aggregate storage databases:

```
BEGIN
DOUBLE
ELSE
END
FUNCTION
GLOBAL
IF
MACRO
MEMBER
RANGE
RETURN
STRING
THEN
```

Glossary

! *See bang character (!).*

#MISSING *See missing data (#MISSING).*

access permissions A set of operations that a user can perform on a resource.

accessor Input and output data specifications for data mining algorithms.

account blocking The process by which accounts accept input data in the consolidated file. Blocked accounts do not receive their value through the additive consolidation process.

account eliminations Accounts which have their values set to zero in the consolidated file during consolidation.

account type How an account's value flows over time, and its sign behavior. Account type options can include expense, income, asset, liability, and equity.

accountability map A visual, hierarchical representation of the responsibility, reporting, and dependency structure of the accountability teams (also known as critical business areas) in an organization.

accounts dimension A dimension type that makes accounting intelligence available. Only one dimension can be defined as Accounts.

active service A service whose Run Type is set to Start rather than Hold.

activity-level authorization Defines user access to applications and the types of activities they can perform on applications, independent of the data that will be operated on.

ad hoc report An online analytical query created on-the-fly by an end user.

adapter Software that enables a program to integrate with data and metadata from target and source systems.

adaptive states Interactive Reporting Web Client level of permission.

adjustment *See journal entry (JE).*

Advanced Relational Access The integration of a relational database with an Essbase multidimensional database so that all data remains in the relational database and is mapped to summary-level data residing in the Essbase database.

agent An Essbase server process that starts and stops applications and databases, manages connections from users, and handles user-access security. The agent is referred to as ESSBASE.EXE.

aggregate cell A cell comprising several cells. For example, a data cell that uses Children(Year) expands to four cells containing Quarter 1, Quarter 2, Quarter 3, and Quarter 4 data.

aggregate function A type of function, such as sum or calculation of an average, that summarizes or performs analysis on data.

aggregate limit A limit placed on an aggregated request line item or aggregated metatopic item.

aggregate storage database The database storage model designed to support large-scale, sparsely distributed data which is categorized into many, potentially large dimensions. Upper level members and formulas are dynamically calculated, and selected data values are aggregated and stored, typically with improvements in overall aggregation time.

aggregate view A collection of aggregate cells based on the levels of the members within each dimension. To reduce calculation time, values are pre-aggregated and stored as aggregate views. Retrievals start from aggregate view totals and add up from there.

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

aggregation script In aggregate storage databases only, a file that defines a selection of aggregate views to be built into an aggregation.

alias An alternative name. For example, for a more easily identifiable column descriptor you can display the alias instead of the member name.

alias table A table that contains alternate names for members.

alternate hierarchy A hierarchy of shared members. An alternate hierarchy is based upon an existing hierarchy in a database outline, but has alternate levels in the dimension. An alternate hierarchy allows the same data to be seen from different points of view.

ancestor A branch member that has members below it. For example, the members Qtr2 and 2006 are ancestors of the member April.

appender A Log4j term for destination.

application (1) A software program designed to run a specific task or group of tasks such as a spreadsheet program or database management system. (2) A related set of dimensions and dimension members that are used to meet a specific set of analytical and/or reporting requirements.

application currency The default reporting currency for the application.

area A predefined set of members and values that makes up a partition.

arithmetic data load A data load that performs operations on values in the database, such as adding 10 to each value.

artifact An individual application or repository item; for example, scripts, forms, rules files, Interactive Reporting documents, and financial reports. Also known as an object.

assemblies Installation files for EPM System products or components.

asset account An account type that stores values that represent a company's assets.

assignment The association of a source and destination in the allocation model that controls the direction of allocated costs or revenue flow within Profitability and Cost Management.

attribute Characteristic of a dimension member. For example, Employee dimension members may have attributes of Name, Age, or Address. Product dimension members can have several attributes, such as a size and flavor.

attribute association A relationship in a database outline whereby a member in an attribute dimension describes a characteristic of a member of its base dimension. For example, if product 100-10 has a grape flavor, the product 100-10 has the Flavor attribute association of grape. Thus, the 100-10 member of the Product dimension is associated with the Grape member of the Flavor attribute dimension.

Attribute Calculations dimension A system-defined dimension that performs these calculation operations on groups of members: Sum, Count, Avg, Min, and Max. This dimension is calculated dynamically and is not visible in the database outline. For example, using the Avg member, you can calculate the average sales value for Red products in New York in January.

attribute dimension A type of dimension that enables analysis based on the attributes or qualities of dimension members.

attribute reporting A reporting process based on the attributes of the base dimension members. *See also base dimension*.

attribute type A text, numeric, Boolean, date, or linked-attribute type that enables different functions for grouping, selecting, or calculating data. For example, because the Ounces attribute dimension has the type numeric, the number of ounces specified as the attribute of each product can be used to calculate the profit per ounce for that product.

authentication Verification of identity as a security measure. Authentication is typically based on a user name and password. Passwords and digital signatures are forms of authentication.

authentication service A core service that manages one authentication system.

auto-reversing journal A journal for entering adjustments that you want to reverse in the next period.

automated stage A stage that does not require human intervention, for example, a data load.

axis (1) A straight line that passes through a graphic used for measurement and categorization. (2) A report aspect used to arrange and relate multidimensional data, such as filters, pages, rows, and columns. For example, for a data query in Simple Basic, an axis can define columns for values for Qtr1, Qtr2, Qtr3, and Qtr4. Row data would be retrieved with totals in the following hierarchy: Market, Product.

backup A duplicate copy of an application instance.

balance account An account type that stores unsigned values that relate to a particular point in time.

balanced journal A journal in which the total debits equal the total credits.

bang character (!) A character that terminates a series of report commands and requests information from the database. A report script must be terminated with a bang character; several bang characters can be used within a report script.

bar chart A chart that can consist of one to 50 data sets, with any number of values assigned to each data set. Data sets are displayed as groups of corresponding bars, stacked bars, or individual bars in separate rows.

base currency The currency in which daily business transactions are performed.

base dimension A standard dimension that is associated with one or more attribute dimensions. For example, assuming products have flavors, the Product dimension is the base dimension for the Flavors attribute dimension.

base entity An entity at the bottom of the organization structure that does not own other entities.

batch calculation Any calculation on a database that is done in batch; for example, a calculation script or a full database calculation. Dynamic calculations are not considered to be batch calculations.

batch file An operating system file that can call multiple ESSCMD scripts and run multiple sessions of ESSCMD. On Windows-based systems, batch files have BAT file extensions. On UNIX, batch files are written as a shell script.

batch loader An FDM component that enables the processing of multiple files.

batch POV A collection of all dimensions on the user POV of every report and book in the batch. While scheduling the batch, you can set the members selected on the batch POV.

batch processing mode A method of using ESSCMD to write a batch or script file that can be used to automate routine server maintenance and diagnostic tasks. ESSCMD script files can execute multiple commands and can be run from the operating system command line or from within operating system batch files. Batch files can be used to call multiple ESSCMD scripts or run multiple instances of ESSCMD.

block The primary storage unit which is a multidimensional array representing the cells of all dense dimensions.

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

Blocked Account An account that you do not want calculated in the consolidated file because you want to enter it manually.

book A container that holds a group of similar Financial Reporting documents. Books may specify dimension sections or dimension changes.

book POV The dimension members for which a book is run.

bookmark A link to a reporting document or a Web site, displayed on a personal page of a user. The two types of bookmarks are My Bookmarks and image bookmarks.

bounding rectangle The required perimeter that encapsulates the Interactive Reporting document content when embedding Interactive Reporting document sections in a personal page, specified in pixels for height and width or row per page.

broadcast message A simple text message sent by an administrator to a user who is logged on to a Planning application. The message displays information to the user such as system availability, notification of application refresh, or application backups.

budget administrator A person responsible for setting up, configuring, maintaining, and controlling an application. Has all application privileges and data access permissions.

build method A method used to modify database outlines. Choice of a build method is based on the format of data in data source files.

business process A set of activities that collectively accomplish a business objective.

business rules Logical expressions or formulas that are created within an application to produce a desired set of resulting values.

cache A buffer in memory that holds data temporarily.

calc script A set of commands that define how a database is consolidated or aggregated. A calculation script may also contain commands that specify allocation and other calculation rules separate from the consolidation process.

calculated member in MaxL DML A member designed for analytical purposes and defined in the optional WITH section of a MaxL DML query.

calculated member in MaxL DML A member designed for analytical purposes and defined in the optional WITH section of a MaxL DML query.

calculation The process of aggregating data, or of running a calculation script on a database.

Calculation Manager A module of Performance Management Architect that Planning and Financial Management users can use to design, validate, and administrate business rules in a graphical environment.

calculation status A consolidation status that indicates that some values or formula calculations have changed. You must reconsolidate to get the correct values for the affected entity.

calendar User-defined time periods and their relationship to each other. Q1, Q2, Q3, and Q4 comprise a calendar or fiscal year.

cascade The process of creating multiple reports for a subset of member values.

Catalog pane Displays a list of elements available to the active section. If Query is the active section, a list of database tables is displayed. If Pivot is the active section, a list of results columns is displayed. If Dashboard is the active section, a list of embeddable sections, graphic tools, and control tools are displayed.

categories Groupings by which data is organized. For example, Month.

cause and effect map Depicts how the elements that form your corporate strategy relate and how they work together to meet your organization's strategic goals. A Cause and Effect map tab is automatically created for each Strategy map.

CDF *See custom-defined function (CDF).*

CDM *See custom-defined macro (CDM).*

cell (1) The data value at the intersection of dimensions in a multidimensional database; the intersection of a row and a column in a worksheet. (2) A logical group of nodes belonging to one administrative domain.

cell note A text annotation for a cell in an Essbase database. Cell notes are a type of LRO.

CHANGED status Consolidation status that indicates data for an entity has changed.

chart A graphical representation of spreadsheet data. The visual nature expedites analysis, color-coding, and visual cues that aid comparisons.

chart template A template that defines the metrics to display in Workspace charts.

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

choice list A list of members that a report designer can specify for each dimension when defining the report's point of view. A user who wants to change the point of view for a dimension that uses a choice list can select only the members specified in that defined member list or those members that meet the criteria defined in the function for the dynamic list.

clean block A data block that where the database is fully calculated, if a calculation script calculates all dimensions at once, or if the SET CLEARUPDATESTATUS command is used in a calculation script.

cluster An array of servers or databases that behave as a single resource which share task loads and provide failover support; eliminates one server or database as a single point of failure in a system.

clustered bar charts Charts in which categories are viewed side-by-side; useful for side-by-side category analysis; used only with vertical bar charts.

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

column A vertical display of information in a grid or table. A column can contain data from one field, derived data from a calculation, or textual information.

committed access An Essbase Kernel Isolation Level setting that affects how Essbase handles transactions. Under committed access, concurrent transactions hold long-term write locks and yield predictable results.

computed item A virtual column (as opposed to a column that is physically stored in the database or cube) that can be calculated by the database during a query, or by Interactive Reporting Studio in the Results section. Computed items are calculations of data based on functions, data items, and operators provided in the dialog box and can be included in reports or reused to calculate other data.

configuration file The security platform relies on XML documents to be configured by the product administrator or software installer. The XML document must be modified to indicate meaningful values for properties, specifying locations and attributes pertaining to the corporate authentication scenario.

connection file *See Interactive Reporting connection file (.oce)*.

consolidated file (Parent) A file into which all of the business unit files are consolidated; contains the definition of the consolidation.

consolidation The process of aggregating data from dependent entities to parent entities. For example, if the dimension Year consists of the members Qtr1, Qtr2, Qtr3, and Qtr4, its consolidation is Year.

consolidation file (*.cns) The consolidation file is a graphical interface that enables you to add, delete or move Strategic Finance files in the consolidation process using either a Chart or Tree view. It also enables you to define and modify the consolidation.

consolidation rule Identifies the rule that is executed during the consolidation of the node of the hierarchy. This rule can contain customer specific formulas appropriate for the correct consolidation of parent balances. Elimination processing can be controlled within these rules.

content Information stored in the repository for any type of file.

content browser A Component that allows users to Browse and select content to be placed in a Workspace Page .

context variable A variable that is defined for a particular task flow to identify the context of the taskflow instance.

contribution The value added to a parent from a child entity. Each child has a contribution to its parent.

controls group Used in FDM to maintain and organize certification and assessment information, especially helpful for meeting Sarbanes-Oxley requirements.

conversion rate *See exchange rate*.

cookie A segment of data placed on your computer by a Web site.

correlated subqueries Subqueries that are evaluated once for every row in the parent query; created by joining a topic item in the subquery with a topic in the parent query.

critical business area (CBA) An individual or a group organized into a division, region, plant, cost center, profit center, project team, or process; also called accountability team or business area.

critical success factor (CSF) A capability that must be established and sustained to achieve a strategic objective; owned by a strategic objective or a critical process and is a parent to one or more actions.

crosstab reporting Categorizes and summarizes data in table format. The table cells contain summaries of the data that fit within the intersecting categories. For example, a crosstab report of product sales information could show size attributes, such as Small and Large, as column headings and color attributes, such as Blue and Yellow, as row headings. The cell in the table where Large and Blue intersect could contain the total sales of all Blue products that are sized Large.

cube A block of data that contains three or more dimensions. An Essbase database is a cube.

cube deployment In Essbase Studio, the process of setting load options for a model to build an outline and load data into an Essbase application and database.

cube schema In Essbase Studio, the metadata elements, such as measures and hierarchies, representing the logical model of a cube.

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

Currency Overrides In any input period, the selected input method can be overridden to enable input of that period's value as Default Currency/Items. To override the input method, enter a pound sign (#) either before or after the number.

currency partition A dimension type that separates local currency members from a base currency, as defined in an application. Identifies currency types, such as Actual, Budget, and Forecast.

custom calendar Any calendar created by an administrator.

custom dimension A dimension created and defined by users. Channel, product, department, project, or region could be custom dimensions.

custom property A property of a dimension or dimension member that is created by a user.

custom report A complex report from the Design Report module, composed of any combination of components.

custom-defined function (CDF) Essbase calculation functions developed in Java and added to the standard Essbase calculation scripting language using MaxL. *See also custom-defined macro (CDM).*

custom-defined macro (CDM) Essbase macros written with Essbase calculator functions and special macro functions. Custom-defined macros use an internal Essbase macro language that enables the combination of calculation functions and they operate on multiple input parameters. *See also custom-defined function (CDF).*

cycle through To perform multiple passes through a database while calculating it.

dashboard A collection of metrics and indicators that provide an interactive summary of your business. Dashboards enable you to build and deploy analytic applications.

data cache A buffer in memory that holds uncompressed data blocks.

data cell *See cell.*

data file cache A buffer in memory that holds compressed data (PAG) files.

data form A grid display that enables users to enter data into the database from an interface such as a Web browser, and to view and analyze data or related text. Certain dimension member values are fixed, giving users a specific view into the data.

data function That computes aggregate values, including averages, maximums, counts, and other statistics, that summarize groupings of data.

data load location In FDM, a reporting unit responsible for submitting source data into the target system. Typically, there is one FDM data load location for each source file loaded to the target system.

data load rules A set of criteria that determines how to load data from a text-based file, a spreadsheet, or a relational data set into a database.

data lock Prevents changes to data according to specified criteria, such as period or scenario.

data mining The process of searching through an Essbase database for hidden relationships and patterns in a large amount of data.

data model A representation of a subset of database tables.

data value *See cell.*

database connection File that stores definitions and properties used to connect to data sources and enables database references to be portable and widely used.

date measure In Essbase, a member tagged as "Date" in the dimension where measures are represented. The cell values are displayed as formatted dates. Dates as measures can be useful for types of analysis that are difficult to represent using the Time dimension. For example, an application may need to track acquisition dates for a series of capital assets, but the acquisition dates span too large a period to allow for feasible Time dimension modeling. *See also typed measure.*

Default Currency Units Define the unit scale of data. For example, if you select to define your analysis in Thousands, and enter "10", this is interpreted as "10,000".

dense dimension In block storage databases, a dimension likely to contain data for every combination of dimension members. For example, time dimensions are often dense because they can contain all combinations of all members. *Contrast with sparse dimension.*

dependent entity An entity that is owned by another entity in the organization.

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

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

Design Report An interface in Web Analysis Studio for designing custom reports, from a library of components.

destination Within a Profitability and Cost Management assignment, the destination is the receiving point for allocated values.

destination currency The currency to which balances are converted. You enter exchange rates and convert from the source currency to the destination currency. For example, when you convert from EUR to USD, the destination currency is USD.

detail chart A chart that provides the detailed information that you see in a Summary chart. Detail charts appear in the Investigate Section in columns below the Summary charts. If the Summary chart shows a Pie chart, then the Detail charts below represent each piece of the pie.

dimension A data category used to organize business data for retrieval and preservation of values. Dimensions usually contain hierarchies of related members grouped within them. For example, a Year dimension often includes members for each time period, such as quarters and months.

dimension build The process of adding dimensions and members to an Essbase outline.

dimension build rules Specifications, similar to data load rules, that Essbase uses to modify an outline. The modification is based on data in an external data source file.

dimension tab In the Pivot section, the tab that enables you to pivot data between rows and columns.

dimension table (1) A table that includes numerous attributes about a specific business process. (2) In Essbase Integration Services, a container in the OLAP model for one or more relational tables that define a potential dimension in Essbase.

dimension type A dimension property that enables the use of predefined functionality. Dimensions tagged as time have a predefined calendar functionality.

dimensionality In MaxL DML, the represented dimensions (and the order in which they are represented) in a set. For example, the following set consists of two tuples of the same dimensionality because they both reflect the dimensions (Region, Year): { (West, Feb), (East, Mar) }

direct rate A currency rate that you enter in the exchange rate table. The direct rate is used for currency conversion. For example, to convert balances from JPY to USD, In the exchange rate table, enter a rate for the period/scenario where the source currency is JPY and the destination currency is USD.

dirty block A data block containing cells that have been changed since the last calculation. Upper level blocks are marked as dirty if their child blocks are dirty (that is, they have been updated).

display type One of three Web Analysis formats saved to the repository: spreadsheet, chart, and pinboard.

dog-ear The flipped page corner in the upper right corner of the chart header area.

domain In data mining, a variable representing a range of navigation within data.

drill-down Navigation through the query result set using the dimensional hierarchy. Drilling down moves the user perspective from aggregated data to detail. For example, drilling down can reveal hierarchical relationships between years and quarters or quarters and months.

drill-through The navigation from a value in one data source to corresponding data in another source.

driver A driver is an allocation method that describes the mathematical relationship between the sources that utilize the driver, and the destinations to which those sources allocate cost or revenue.

duplicate alias name A name that occurs more than once in an alias table and that can be associated with more than one member in a database outline. Duplicate alias names can be used with duplicate member outlines only.

duplicate member name The multiple occurrence of a member name in a database, with each occurrence representing a different member. For example, a database has two members named "New York." One member represents New York state and the other member represents New York city.

duplicate member outline A database outline containing duplicate member names.

Dynamic Calc and Store members A member in a block storage outline that Essbase calculates only upon the first retrieval of the value. Essbase then stores the calculated value in the database. Subsequent retrievals do not require calculating.

Dynamic Calc members A member in a block storage outline that Essbase calculates only at retrieval time. Essbase discards calculated values after completing the retrieval request.

dynamic calculation In Essbase, a calculation that occurs only when you retrieve data on a member that is tagged as Dynamic Calc or Dynamic Calc and Store. The member's values are calculated at retrieval time instead of being precalculated during batch calculation.

dynamic hierarchy In aggregate storage database outlines only, a hierarchy in which members are calculated at retrieval time.

dynamic member list A system-created named member set that is based on user-defined criteria. The list is refreshed automatically whenever it is referenced in the application. As dimension members are added and deleted, the list automatically reapplies the criteria to reflect the changes.

dynamic reference A pointer in the rules file to header records in a data source.

dynamic report A report containing data that is updated when you run the report.

Dynamic Time Series A process that performs period-to-date reporting in block storage databases.

dynamic view account An account type indicating that account values are calculated dynamically from the data that is displayed.

Eliminated Account An account that does not appear in the consolidated file.

elimination The process of zeroing out (eliminating) transactions between entities within an organization.

employee A user responsible for, or associated with, specific business objects. Employees need not work for an organization; for example, they can be consultants. Employees must be associated with user accounts for authorization purposes.

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

ending period A period enabling you to adjust the date range in a chart. For example, an ending period of "month", produces a chart showing information through the end of the current month.

Enterprise View An Administration Services feature that enables management of the Essbase environment from a graphical tree view. From Enterprise View, you can operate directly on Essbase artifacts.

entity A dimension representing organizational units. Examples: divisions, subsidiaries, plants, regions, products, or other financial reporting units.

Equity Beta The riskiness of a stock, measured by the variance between its return and the market return, indicated by an index called "beta". For example, if a stock's return normally moves up or down 1.2% when the market moves up or down 1%, the stock has a beta of 1.2.

essbase.cfg An optional configuration file for Essbase. Administrators may edit this file to customize Essbase Server functionality. Some configuration settings may also be used with Essbase clients to override Essbase Server settings.

EssCell A function entered into a cell in Essbase Spreadsheet Add-in to retrieve a value representing an intersection of specific Essbase database members.

ESSCMD A command-line interface for performing Essbase operations interactively or through batch script files.

ESSLANG The Essbase environment variable that defines the encoding used to interpret text characters. *See also encoding*.

ESSMSH *See MaxL Shell*.

exceptions Values that satisfy predefined conditions. You can define formatting indicators or notify subscribing users when exceptions are generated.

exchange rate A numeric value for converting one currency to another. For example, to convert 1 USD into EUR, the exchange rate of 0.8936 is multiplied with the U.S. dollar. The European euro equivalent of $1 is 0.8936.

exchange rate type An identifier for an exchange rate. Different rate types are used because there may be multiple rates for a period and year. Users traditionally define rates at period end for the average rate of the period and for the end of the period. Additional rate types are historical rates, budget rates, forecast rates, and so on. A rate type applies to one point in time.

expense account An account that stores periodic and year-to-date values that decrease net worth if they are positive.

Extensible Markup Language (XML) A language comprising a set of tags used to assign attributes to data that can be interpreted between applications according to a schema.

external authentication Logging on to Oracle's Hyperion applications with user information stored outside the applications, typically in a corporate directory such as MSAD or NTLM.

externally triggered events Non-time-based events for scheduling job runs.

Extract, Transform, and Load (ETL) Data source-specific programs for extracting data and migrating it to applications.

extraction command An Essbase reporting command that handles the selection, orientation, grouping, and ordering of raw data extracted from a database; begins with the less than (<) character.

fact table The central table in a star join schema, characterized by a foreign key and elements drawn from a dimension table. This table typically contains numeric data that can be related to all other tables in the schema.

Favorites gadget Contains links to Reporting and Analysis documents and URLs.

field An item in a data source file to be loaded into an Essbase database.

file delimiter Characters, such as commas or tabs, that separate fields in a data source.

filter A constraint on data sets that restricts values to specific criteria; for example, to exclude certain tables, metadata, or values, or to control access.

flow account An unsigned account that stores periodic and year-to-date values.

folder A file containing other files for the purpose of structuring a hierarchy.

footer Text or images at the bottom of report pages, containing dynamic functions or static text such as page numbers, dates, logos, titles or file names, and author names.

format Visual characteristics of documents or report objects.

format string In Essbase, a method for transforming the way cell values are displayed.

formula A combination of operators, functions, dimension and member names, and numeric constants calculating database members.

frame An area on the desktop. There are two main areas: the navigation and Workspace frames.

free-form grid An object for presenting, entering, and integrating data from different sources for dynamic calculations.

free-form reporting Creating reports by entering dimension members or report script commands in worksheets.

function A routine that returns values or database members.

gadget Simple, specialized, lightweight applications that provide easy viewing of EPM content and enable access to core Reporting and Analysis functionality.

genealogy data Additional data that is optionally generated after allocation calculations. This data enables reporting on all cost or revenue flows from start to finish through all allocation steps.

generation A layer in a hierarchical tree structure that defines member relationships in a database. Generations are ordered incrementally from the top member of the dimension (generation 1) down to the child members. Use the unique generation name to identify a layer in the hierarchical tree structure.

generic jobs Non-SQR Production Reporting or non-Interactive Reporting jobs.

global report command A command in a running report script that is effective until replaced by another global command or the file ends.

grid POV A means for specifying dimension members on a grid without placing dimensions in rows, columns, or page intersections. A report designer can set POV values at the grid level, preventing user POVs from affecting the grid. If a dimension has one grid value, you put the dimension into the grid POV instead of the row, column, or page.

group A container for assigning similar access permissions to multiple users.

GUI Graphical user interface

head up display A mode that shows your loaded Smart Space desktop including the background image above your Windows desktop.

highlighting Depending on your configuration, chart cells or ZoomChart details may be highlighted, indicating value status: red (bad), yellow (warning), or green (good).

Historical Average An average for an account over a number of historical periods.

holding company An entity that is part of a legal entity group, with direct or indirect investments in all entities in the group.

host A server on which applications and services are installed.

host properties Properties pertaining to a host, or if the host has multiple Install_Homes, to an Install_Home. The host properties are configured from the CMC.

Hybrid Analysis An analysis mapping low-level data stored in a relational database to summary-level data stored in Essbase, combining the mass scalability of relational systems with multidimensional data.

hyperlink A link to a file, Web page, or an intranet HTML page.

Hypertext Markup Language (HTML) A programming language specifying how Web browsers display data.

identity A unique identification for a user or group in external authentication.

image bookmarks Graphic links to Web pages or repository items.

IMPACTED status Indicates changes in child entities consolidating into parent entities.

implied share A member with one or more children, but only one is consolidated, so the parent and child share a value.

import format In FDM, defines the structure of the source file which enables the loading of a source data file to an FDM data load location.

inactive group A group for which an administrator has deactivated system access.

inactive service A service suspended from operating.

INACTIVE status Indicates entities deactivated from consolidation for the current period.

inactive user A user whose account has been deactivated by an administrator.

income account An account storing periodic and year-to-date values that, if positive, increase net worth.

index (1) A method where Essbase uses sparse-data combinations to retrieve data in block storage databases. (2) The index file.

index cache A buffer containing index pages.

index entry A pointer to an intersection of sparse dimensions. Index entries point to data blocks on disk and use offsets to locate cells.

index file An Essbase file storing block storage data retrieval information, residing on disk, and containing index pages.

index page A subdivision in an index file. Contains pointers to data blocks.

input data Data loaded from a source rather than calculated.

Install_Home A variable for the directory where EPM System products are installed. Refers to one instance of an EPM System product when multiple applications are installed on the same computer.

integration Process that is run to move data between EPM System products using Shared Services. Data integration definitions specify the data moving between a source application and a destination application, and enable the data movements to be grouped, ordered, and scheduled.

intelligent calculation A calculation method tracking updated data blocks since the last calculation.

Interactive Reporting connection file (.oce) Files encapsulating database connection information, including: the database API (ODBC, SQL*Net, etc.), database software, the database server network address, and database user name. Administrators create and publish Interactive Reporting connection files (.oce).

intercompany elimination *See elimination.*

intercompany matching The process of comparing balances for pairs of intercompany accounts within an application. Intercompany receivables are compared to intercompany payables for matches. Matching accounts are used to eliminate intercompany transactions from an organization's consolidated totals.

intercompany matching report A report that compares intercompany account balances and indicates if the accounts are in, or out, of balance.

interdimensional irrelevance A situation in which a dimension does not intersect with other dimensions. Because the data in the dimension cannot be accessed from the non-intersecting dimensions, the non-intersecting dimensions are not relevant to that dimension.

intersection A unit of data representing the intersection of dimensions in a multidimensional database; also, a worksheet cell.

intrastage assignment Assignments in the financial flow that are assigned to objects within the same stage.

introspection A deep inspection of a data source to discover hierarchies based on the inherent relationships in the database. *Contrast with scraping.*

Investigation *See drill-through.*

isolation level An Essbase Kernel setting that determines the lock and commit behavior of database operations. Choices are: committed access and uncommitted access.

iteration A "pass" of the budget or planning cycle in which the same version of data is revised and promoted.

Java Database Connectivity (JDBC) A client-server communication protocol used by Java based clients and relational databases. The JDBC interface provides a call-level API for SQL-based database access.

job output Files or reports produced from running a job.

jobs Documents with special properties that can be launched to generate output. A job can contain Interactive Reporting, SQR Production Reporting, or generic documents.

join A link between two relational database tables or topics based on common content in a column or row. A join typically occurs between identical or similar items within different tables or topics. For example, a record in the Customer table is joined to a record in the Orders table because the Customer ID value is the same in each table.

journal entry (JE) A set of debit/credit adjustments to account balances for a scenario and period.

JSP Java Server Pages.

KeyContacts gadget Contains a group of Smart Space users and provides access to Smart Space Collaborator. For example, you can have a KeyContacts gadget for your marketing team and another for your development team.

latest A Spreadsheet key word used to extract data values from the member defined as the latest time period.

layer (1) The horizontal location of members in a hierarchical structure, specified by generation (top down) or level (bottom up). (2) Position of objects relative to other objects. For example, in the Sample Basic database, Qtr1 and Qtr4 are in the same layer, so they are also in the same generation, but in a database with a ragged hierarchy, Qtr1 and Qtr4 might not be in same layer, though they are in the same generation.

layout area Used to designate an area on a Workspace Page where content can be placed.

legend box A box containing labels that identify the data categories of a dimension.

level A layer in a hierarchical tree structure that defines database member relationships. Levels are ordered from the bottom dimension member (level 0) up to the parent members.

level 0 block A data block for combinations of sparse, level 0 members.

level 0 member A member that has no children.

liability account An account type that stores “point in time” balances of a company's liabilities. Examples of liability accounts include accrued expenses, accounts payable, and long term debt.

life cycle management The process of managing application information from inception to retirement.

Lifecycle Management Utility A command-line utility for migrating applications and artifacts.

line chart A chart that displays one to 50 data sets, each represented by a line. A line chart can display each line stacked on the preceding ones, as represented by an absolute value or a percent.

line item detail The lowest level of detail in an account.

lineage The relationship between different metadata elements showing how one metadata element is derived from one or more other metadata elements, ultimately tracing the metadata element to its physical source. In Essbase Studio, a lineage viewer displays the relationships graphically. *See also traceability*.

link (1) A reference to a repository object. Links can reference folders, files, shortcuts, and other links. (2) In a task flow, the point where the activity in one stage ends and another begins.

link condition A logical expression evaluated by the taskflow engine to determine the sequence of launching taskflow stages.

linked data model Documents that are linked to a master copy in a repository.

linked partition A shared partition that enables you to use a data cell to link two databases. When a user clicks a linked cell in a worksheet, Essbase opens a new sheet displaying the dimensions in the linked database. The user can then drill down those dimensions.

linked reporting object (LRO) A cell-based link to an external file such as cell notes, URLs, or files with text, audio, video, or pictures. (Only cell notes are supported for Essbase LROs in Financial Reporting.) *Contrast with local report object*.

local currency An input currency type. When an input currency type is not specified, the local currency matches the entity's base currency.

local report object A report object that is not linked to a Financial Reporting report object in Explorer. *Contrast with linked reporting object (LRO)*.

local results A data model's query results. Results can be used in local joins by dragging them into the data model. Local results are displayed in the catalog when requested.

locale A computer setting that specifies a location's language, currency and date formatting, data sort order, and the character set encoding used on the computer. Essbase uses only the encoding portion. *See also encoding* and *ESSLANG*.

locale header record A text record at the beginning of some non-Unicode-encoded text files, such as scripts, that identifies the encoding locale.

location alias A descriptor that identifies a data source. The location alias specifies a server, application, database, user name, and password. Location aliases are set by DBAs at the database level using Administration Services Console, ESSCMD, or the API.

locked A user-invoked process that prevents users and processes from modifying data.

locked data model Data models that cannot be modified by a user.

LOCKED status A consolidation status indicating that an entity contains data that cannot be modified.

Log Analyzer An Administration Services feature that enables filtering, searching, and analysis of Essbase logs.

logic group In FDM, contains one or more logic accounts that are generated after a source file is loaded into FDM. Logic accounts are calculated accounts that are derived from the source data.

LRO *See linked reporting object (LRO)*.

managed server An application server process running in its own Java Virtual Machine (JVM).

manual stage A stage that requires human intervention to complete.

Map File Used to store the definition for sending data to or retrieving data from an external database. Map files have different extensions (.mps to send data; .mpr to retrieve data).

Map Navigator A feature that displays your current position on a Strategy, Accountability, or Cause and Effect map, indicated by a red outline.

Marginal Tax Rate Used to calculate the after-tax cost of debt. Represents the tax rate applied to the last earned income dollar (the rate from the highest tax bracket into which income falls) and includes federal, state and local taxes. Based on current level of taxable income and tax bracket, you can predict marginal tax rate.

Market Risk Premium The additional rate of return paid over the risk-free rate to persuade investors to hold "riskier" investments than government securities. Calculated by subtracting the risk-free rate from the expected market return. These figures should closely model future market conditions.

master data model An independent data model that is referenced as a source by multiple queries. When used, "Locked Data Model" is displayed in the Query section's Content pane; the data model is linked to the master data model displayed in the Data Model section, which an administrator may hide.

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

MaxL The multidimensional database access language for Essbase, consisting of a data definition language (MaxL DDL) and a data manipulation language (MaxL DML). *See also MaxL DDL*, *MaxL DML*, and *MaxL Shell*.

MaxL DDL Data definition language used by Essbase for batch or interactive system-administration tasks.

MaxL DML Data manipulation language used in Essbase for data query and extraction.

MaxL Perl Module A Perl module (essbase.pm) that is part of Essbase MaxL DDL. This module can be added to the Perl package to provide access to Essbase databases from Perl programs.

MaxL Script Editor A script-development environment in Administration Services Console. MaxL Script Editor is an alternative to using a text editor and the MaxL Shell for administering Essbase with MaxL scripts.

MaxL Shell An interface for passing MaxL statements to Essbase Server. The MaxL Shell executable file is located in the Essbase bin directory (UNIX: essmsh, Windows: essmsh.exe).

MDX (multidimensional expression) The language that give instructions to OLE DB for OLAP- compliant databases, as SQL is used for relational databases. When you build the OLAPQuery section's Outliner, Interactive Reporting Clients translate requests into MDX instructions. When you process the query, MDX is sent to the database server, which returns records that answer your query. *See also SQL spreadsheet.*

measures Numeric values in an OLAP database cube that are available for analysis. Measures are margin, cost of goods sold, unit sales, budget amount, and so on. *See also fact table.*

member A discrete component within a dimension. A member identifies and differentiates the organization of similar units. For example, a time dimension might include such members as Jan, Feb, and Qtr1.

member list A named group, system- or user-defined, that references members, functions, or member lists within a dimension.

member load In 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.

merge A data load option that clears values only from the accounts specified in the data load file and replaces them with values in the data load file.

metadata A set of data that defines and describes the properties and attributes of the data stored in a database or used by an application. Examples of metadata are dimension names, member names, properties, time periods, and security.

metadata elements Metadata derived from data sources and other metadata that is stored and cataloged for Essbase Studio use.

metadata sampling The process of retrieving a sample of members in a dimension in a drill-down operation.

metadata security Security set at the member level to restrict users from accessing certain outline members.

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

metric A numeric measurement computed from business data to help assess business performance and analyze company trends.

migration The process of copying applications, artifacts, or users from one environment or computer to another; for example, from a testing environment to a production environment.

migration audit report A report generated from the migration log that provides tracking information for an application migration.

migration definition file (.mdf) A file that contains migration parameters for an application migration, enabling batch script processing.

migration log A log file that captures all application migration actions and messages.

migration snapshot A snapshot of an application migration that is captured in the migration log.

MIME Type (Multipurpose Internet Mail Extension) An attribute that describes the data format of an item, so that the system knows which application should open the object. A file's mime type is determined by the file extension or HTTP header. Plug-ins tell browsers what mime types they support and what file extensions correspond to each mime type.

mining attribute In data mining, a class of values used as a factor in analysis of a set of data.

minireport A report component that includes layout, content, hyperlinks, and the query or queries to load the report. Each report can include one or more minireports.

minischema A graphical representation of a subset of tables from a data source that represents a data modeling context.

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

model (1) In data mining, a collection of an algorithm's findings about examined data. A model can be applied against a wider data set to generate useful information about that data. (2) A file or content string containing an application-specific representation of data. Models are the basic data managed by Shared Services, of two major types: dimensional and non-dimensional application objects. (3) In Business Modeling, a network of boxes connected to represent and calculate the operational and financial flow through the area being examined.

monetary A money-related value.

multidimensional database A method of organizing, storing, and referencing data through three or more dimensions. An individual value is the intersection point for a set of dimensions. *Contrast with relational database*.

multiload An FDM feature that allows the simultaneous loading of multiple periods, categories, and locations.

My Workspace Page A page created with content from multiple sources including documents, URL, and other content types. Enables a user to aggregate content from Oracle and non-Oracle sources.

named set In MaxL DML, a set with its logic defined in the optional WITH section of a MaxL DML query. The named set can be referenced multiple times in the query.

native authentication The process of authenticating a user name and password from within the server or application.

nested column headings A report column heading format that displays data from multiple dimensions. For example, a column heading that contains Year and Scenario members is a nested column. The nested column heading shows Q1 (from the Year dimension) in the top line of the heading, qualified by Actual and Budget (from the Scenario dimension) in the bottom line of the heading.

NO DATA status A consolidation status indicating that this entity contains no data for the specified period and account.

non-dimensional model A Shared Services model type that includes application objects such as security files, member lists, calculation scripts, and Web forms.

non-unique member name *See duplicate member name*.

note Additional information associated with a box, measure, scorecard or map element.

Notifications gadget Shows notification message history received from other users or systems.

null value A value that is absent of data. Null values are not equal to zero.

numeric attribute range A feature used to associate a base dimension member that has a discrete numeric value with an attribute that represents a value range. For example, to classify customers by age, an Age Group attribute dimension can contain members for the following age ranges: 0-20, 21-40, 41-60, and 61-80. Each Customer dimension member can be associated with an Age Group range. Data can be retrieved based on the age ranges rather than on individual age values.

ODBC Open Database Connectivity. A database access method used from any application regardless of how the database management system (DBMS) processes the information.

OK status A consolidation status indicating that an entity has already been consolidated, and that data has not changed below it in the organization structure.

OLAP Metadata Catalog In Integration Services, a relational database containing metadata describing the nature, source, location, and type of data that is pulled from the relational data source.

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

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

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

organization An entity hierarchy that defines each entity and their relationship to others in the hierarchy.

origin The intersection of two axes.

outline The database structure of a multidimensional database, including all dimensions, members, tags, types, consolidations, and mathematical relationships. Data is stored in the database according to the structure defined in the outline.

outline synchronization For partitioned databases, the process of propagating outline changes from one database to another database.

P&L accounts (P&L) Profit and loss accounts. Refers to a typical grouping of expense and income accounts that comprise a company's income statement.

page A display of information in a grid or table often represented by the Z-axis. A page can contain data from one field, derived data from a calculation, or text.

page file Essbase data file.

page heading A report heading type that lists members represented on the current page of the report. All data values on the page have the members in the page heading as a common attribute.

page member A member that determines the page axis.

palette A JASC compliant file with a .PAL extension. Each palette contains 16 colors that complement each other and can be used to set the dashboard color elements.

parallel calculation A calculation option. Essbase divides a calculation into tasks and calculates some tasks simultaneously.

parallel data load In Essbase, the concurrent execution of data load stages by multiple process threads.

parallel export The ability to export Essbase data to multiple files. This may be faster than exporting to a single file, and it may resolve problems caused by a single data file becoming too large for the operating system to handle.

parent adjustments The journal entries that are posted to a child in relation to its parent.

parents The entities that contain one or more dependent entities that report directly to them. Because parents are both entities and associated with at least one node, they have entity, node, and parent information associated with them.

partition area A sub cube within a database. A partition is composed of one or more areas of cells from a portion of the database. For replicated and transparent partitions, the number of cells within an area must be the same for the data source and target to ensure that the two partitions have the same shape. If the data source area contains 18 cells, the data target area must also contain 18 cells to accommodate the number of values.

partitioning The process of defining areas of data that are shared or linked between data models. Partitioning can affect the performance and scalability of Essbase applications.

pattern matching The ability to match a value with any or all characters of an item entered as a criterion. Missing characters may be represented by wild card values such as a question mark (?) or an asterisk (*). For example, “Find all instances of apple” returns apple, but “Find all instances of apple*” returns apple, applesauce, applecranberry, and so on.

percent consolidation The portion of a child's values that is consolidated to its parent.

percent control Identifies the extent to which an entity is controlled within the context of its group.

percent ownership Identifies the extent to which an entity is owned by its parent.

performance indicator An image file used to represent measure and scorecard performance based on a range you specify; also called a status symbol. You can use the default performance indicators or create an unlimited number of your own.

periodic value method (PVA) A process of currency conversion that applies the periodic exchange rate values over time to derive converted results.

permission A level of access granted to users and groups for managing data or other users and groups.

persistence The continuance or longevity of effect for any Essbase operation or setting. For example, an Essbase administrator may limit the persistence of user name and password validity.

personal pages A personal window to repository information. You select what information to display and its layout and colors.

personal recurring time events Reusable time events that are accessible only to the user who created them.

personal variable A named selection statement of complex member selections.

perspective A category used to group measures on a scorecard or strategic objectives within an application. A perspective can represent a key stakeholder (such as a customer, employee, or shareholder/financial) or a key competency area (such as time, cost, or quality).

pie chart A chart that shows one data set segmented in a pie formation.

pinboard One of the three data object display types. Pinboards are graphics, composed of backgrounds and interactive icons called pins. Pinboards require traffic lighting definitions.

pins Interactive icons placed on graphic reports called pinboards. Pins are dynamic. They can change images and traffic lighting color based on the underlying data values and analysis tools criteria.

pivot The ability to alter the perspective of retrieved data. When Essbase first retrieves a dimension, it expands data into rows. You can then pivot or rearrange the data to obtain a different viewpoint.

planner Planners, who comprise the majority of users, can input and submit data, use reports that others create, execute business rules, use task lists, enable e-mail notification for themselves, and use Smart View.

planning unit A data slice at the intersection of a scenario, version, and entity; the basic unit for preparing, reviewing, annotating, and approving plan data.

plot area The area bounded by X, Y, and Z axes; for pie charts, the rectangular area surrounding the pie.

plug account An account in which the system stores any out of balance differences between intercompany account pairs during the elimination process.

post stage assignment Assignments in the allocation model that are assigned to locations in a subsequent model stage.

POV (point of view) A feature for setting data focus by selecting members that are not already assigned to row, column, or page axes. For example, selectable POVs in FDM could include location, period, category, and target category. In another example, using POV as a filter in Smart View, you could assign the Currency dimension to the POV and select the Euro member. Selecting this POV in data forms displays data in Euro values.

precalculation Calculating the database prior to user retrieval.

precision Number of decimal places displayed in numbers.

predefined drill paths Paths used to drill to the next level of detail, as defined in the data model.

presentation A playlist of Web Analysis documents, enabling reports to be grouped, organized, ordered, distributed, and reviewed. Includes pointers referencing reports in the repository.

preserve formulas User-created formulas kept within a worksheet while retrieving data.

primary measure A high-priority measure important to your company and business needs. Displayed in the Contents frame.

process monitor report Displays a list of locations and their positions within the FDM data conversion process. You can use the process monitor report to monitor the status of the closing process. The report is time-stamped. Therefore, it can be used to determine to which locations at which time data was loaded.

product In Shared Services, an application type, such as Planning or Performance Scorecard.

Production Reporting *See SQR Production Reporting.*

project An instance of EPM System products grouped together in an implementation. For example, a Planning project may consist of a Planning application, an Essbase cube, and a Financial Reporting server instance.

property A characteristic of an artifact, such as size, type, or processing instructions.

provisioning The process of granting users and groups specific access permissions to resources.

proxy server A server acting as an intermediary between workstation users and the Internet to ensure security.

public job parameters Reusable, named job parameters created by administrators and accessible to users with requisite access privileges.

public recurring time events Reusable time events created by administrators and accessible through the access control system.

PVA *See periodic value method (PVA).*

qualified name A member name in a qualified format that differentiates duplicate member names in a duplicate member outline. For example, [Market].[East].[State].[New York] or [Market].[East].[City].[New York]

query Information requests from data providers. For example, used to access relational data sources.

query governor An Essbase Integration server parameter or Essbase server configuration setting that controls the duration and size of queries made to data sources.

range A set of values including upper and lower limits, and values falling between limits. Can contain numbers, amounts, or dates.

reciprocal assignment An assignment in the financial flow that also has the source as one of its destinations.

reconfigure URL URL used to reload servlet configuration settings dynamically when users are already logged on to the Workspace.

record In a database, a group of fields making up one complete entry. For example, a customer record may contain fields for name, address, telephone number, and sales data.

recurring template A journal template for making identical adjustments in every period.

recurring time event An event specifying a starting point and the frequency for running a job.

redundant data Duplicate data blocks that Essbase retains during transactions until Essbase commits updated blocks.

regular journal A feature for entering one-time adjustments for a period. Can be balanced, balanced by entity, or unbalanced.

Related Accounts The account structure groups all main and related accounts under the same main account number. The main account is distinguished from related accounts by the first suffix of the account number.

relational database A type of database that stores data in related two-dimensional tables. *Contrast with multidimensional database.*

replace A data load option that clears existing values from all accounts for periods specified in the data load file, and loads values from the data load file. If an account is not specified in the load file, its values for the specified periods are cleared.

replicated partition A portion of a database, defined through Partition Manager, used to propagate an update to data mastered at one site to a copy of data stored at another site. Users can access the data as though it were part of their local database.

Report Extractor An Essbase component that retrieves report data from the Essbase database when report scripts are run.

report object In report designs, a basic element with properties defining behavior or appearance, such as text boxes, grids, images, and charts.

report script A text file containing Essbase Report Writer commands that generate one or more production reports.

Report Viewer An Essbase component that displays complete reports after report scripts are run.

reporting currency The currency used to prepare financial statements, and converted from local currencies to reporting currencies.

repository Stores metadata, formatting, and annotation information for views and queries.

resources Objects or services managed by the system, such as roles, users, groups, files, and jobs.

restore An operation to reload data and structural information after a database has been damaged or destroyed, typically performed after shutting down and restarting the database.

restructure An operation to regenerate or rebuild the database index and, in some cases, data files.

result frequency The algorithm used to create a set of dates to collect and display results.

review level A Process Management review status indicator representing the process unit level, such as Not Started, First Pass, Submitted, Approved, and Published.

Risk Free Rate The rate of return expected from "safer" investments such as long-term U.S. government securities.

role The means by which access permissions are granted to users and groups for resources.

roll-up *See consolidation*.

root member The highest member in a dimension branch.

RSC services Services that are configured with Remote Service Configurator, including Repository Service, Service Broker, Name Service, Event Service, and Job Service.

runtime prompt A variable that users enter or select before a business rule is run.

sampling The process of selecting a representative portion of an entity to determine the entity's characteristics. *See also metadata sampling*.

saved assumptions User-defined Planning assumptions that drive key business calculations (for example, the cost per square foot of office floor space).

scaling Scaling determines the display of values in whole numbers, tens, hundreds, thousands, millions, and so on.

scenario A dimension for classifying data (for example, Actuals, Budget, Forecast1, and Forecast2).

scope The area of data encompassed by any Essbase operation or setting; for example, the area of data affected by a security setting. Most commonly, scope refers to three levels of granularity, where higher levels encompass lower levels. From highest to lowest, these levels are as follows: the entire system (Essbase Server), applications on Essbase servers, or databases within Essbase server applications. *See also persistence*.

score The level at which targets are achieved, usually expressed as a percentage of the target.

scorecard Business object that represents the progress of an employee, strategy element, or accountability element toward goals. Scorecards ascertain this progress based on data collected for each measure and child scorecard added to the scorecard.

scraping An inspection of a data source to derive the most basic metadata elements from it. *Contrast with introspection*.

Search gadget Searches the Reporting and Analysis repository. The Search gadget looks for a match in the document keywords and description, which are set when you import a document.

secondary measure A low-priority measure, less important than primary measures. Secondary measures do not have Performance reports but can be used on scorecards and to create dimension measure templates.

security agent A Web access management provider (for example, Netegrity SiteMinder) that protects corporate Web resources.

security platform A framework enabling EPM System products to use external authentication and single sign-on.

serial calculation The default calculation setting. Divides a calculation pass into tasks and calculates one task at a time.

services Resources that enable business items to be retrieved, changed, added, or deleted. Examples: Authorization and Authentication.

servlet A piece of compiled code executable by a Web server.

Servlet Configurator A utility for configuring all locally installed servlets.

shared member A member that shares storage space with another member of the same name, preventing duplicate calculation of members that occur multiple times in an Essbase outline.

Shared Services Registry Part of the Shared Services database, the Shared Services Registry stores and re-uses information for most installed EPM System products, including installation directories, database settings, deployment settings, computer names, ports, servers, URLs, and dependent service data.

Shared Workspace Page Workspace Pages shared across an organization which are stored in a special System folder and can be accessed by authorized users from the Shared Workspace Pages Navigate menu.

sibling A child member at the same generation as another child member and having the same immediate parent. For example, the members Florida and New York are children of East and each other's siblings.

single sign-on Ability to access multiple EPM System products after a single login using external credentials.

smart slice In Smart View, a reusable perspective of a data source that contains a restricted set of dimensions or dimension members.

Smart Space client software Runs on the client's computer and provides gadgets, instant collaboration and access to the Reporting and Analysis repository. It is composed of the Smart Space framework and gadgets.

Smart Space Collaborator A service that enables users or systems to send messages and share Reporting and Analysis repository content. The message can take many forms, including instant message style discussions, meetings, and toast messages.

smart tags Keywords in Microsoft Office applications that are associated with predefined actions available from the Smart Tag menu. In EPM System products, smart tags can also be used to import Reporting and Analysis content, and access Financial Management and Essbase functions.

SmartBook gadget Contains documents from the Reporting and Analysis repository or URLs. All documents are loaded when the SmartBook is opened so you can access all content immediately.

SmartCut A link to a repository item, in URL form.

snapshot Read-only data from a specific time.

source currency The currency from which values originate and are converted through exchange rates to the destination currency.

sparse dimension In block storage databases, a dimension unlikely to contain data for all member combinations when compared to other dimensions. For example, not all customers have data for all products. *Contrast with dense dimension*.

SPF files Printer-independent files created by an SQR Production Reporting server, containing a representation of the actual formatted report output, including fonts, spacing, headers, footers, and so on.

Spotlighter A tool that enables color coding based on selected conditions.

SQL spreadsheet A data object that displays the result set of a SQL query.

SQR Production Reporting A specialized programming language for data access, data manipulation, and creating SQR Production Reporting documents.

stage A task description that forms one logical step within a taskflow, usually performed by an individual. A stage can be manual or automated.

stage action For automated stages, the invoked action that executes the stage.

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

standard dimension A dimension that is not an attribute dimension.

standard journal template A journal function used to post adjustments that have common adjustment information for each period. For example, you can create a standard template that contains the common account IDs, entity IDs, or amounts, then use the template as the basis for many regular journals.

Status bar The status bar at the bottom of the screen displays helpful information about commands, accounts, and the current status of your data file.

stored hierarchy In aggregate storage databases outlines only. A hierarchy in which the members are aggregated according to the outline structure. Stored hierarchy members have certain restrictions, for example, they cannot contain formulas.

strategic objective (SO) A long-term goal defined by measurable results. Each strategic objective is associated with one perspective in the application, has one parent, the entity, and is a parent to critical success factors or other strategic objectives.

Strategy map Represents how the organization implements high-level mission and vision statements into lower-level, constituent strategic goals and objectives.

structure view Displays a topic as a simple list of component data items.

Structured Query Language A language used to process instructions to relational databases.

Subaccount Numbering A system for numbering subaccounts using non-sequential, whole numbers.

subscribe Flags an item or folder to receive automatic notification whenever the item or folder is updated.

Summary chart In the Investigates Section, rolls up detail charts shown below in the same column, plotting metrics at the summary level at the top of each chart column.

super service A special service used by the startCommonServices script to start the RSC services.

supervisor A user with full access to all applications, databases, related files, and security mechanisms for a server.

supporting detail Calculations and assumptions from which the values of cells are derived.

suppress rows Excludes rows containing missing values, and underscores characters from spreadsheet reports.

symmetric multiprocessing (SMP) A server architecture that enables multiprocessing and multithreading. Performance is not significantly degraded when a large number of users connect to an single instance simultaneously.

sync Synchronizes Shared Services and application models.

synchronized The condition that exists when the latest version of a model resides in both the application and in Shared Services. *See also model*.

system extract Transfers data from an application's metadata into an ASCII file.

tabs Navigable views of accounts and reports in Strategic Finance.

target Expected results of a measure for a specified period of time (day, quarter, and so on).

task list A detailed status list of tasks for a particular user.

taskflow The automation of a business process in which tasks are passed from one taskflow participant to another according to procedural rules.

taskflow definition Represents business processes in the taskflow management system. Consists of a network of stages and their relationships; criteria indicating the start and end of the taskflow; and information about individual stages, such as participants, associated applications, associated activities, and so on.

taskflow instance Represents a single instance of a taskflow including its state and associated data.

taskflow management system Defines, creates, and manages the execution of a taskflow including: definitions, user or application interactions, and application executables.

taskflow participant The resource who performs the task associated with the taskflow stage instance for both manual and automated stages.

Taxes - Initial Balances Strategic Finance assumes that the Initial Loss Balance, Initial Gain Balance and the Initial Balance of Taxes Paid entries have taken place in the period before the first Strategic Finance time period.

TCP/IP *See Transmission Control Protocol/Internet Protocol (TCP/IP)*.

template A predefined format designed to retrieve particular data consistently.

text list In Essbase, an object that stores text values mapped to numeric identifiers. Text Lists enable the use of text measures.

text measure A data type that allows measure values to be expressed as text. In Essbase, a member tagged as “Text” in the dimension where measures are represented. The cell values are displayed as predefined text. For example, the text measure "Satisfaction Index" may have the values Low, Medium, and High. *See also typed measure*, *text list*, *derived text measure*.

time dimension Defines the time period that the data represents, such as fiscal or calendar periods.

time events Triggers for execution of jobs.

time line viewer An FDM feature that allows a user to view dates and times of completed process flow steps for specific locations.

time scale Displays metrics by a specific period in time, such as monthly or quarterly.

time series reporting A process for reporting data based on a calendar date (for example, year, quarter, month, or week).

Title bar Displays the Strategic Finance name, the file name, and the scenario name Version box.

toast message Messages that appear in the lower right corner of the screen and fade in and out.

token An encrypted identification of one valid user or group on an external authentication system.

top and side labels Column and row headings on the top and sides of a Pivot report.

top-level member A dimension member at the top of the tree in a dimension outline hierarchy, or the first member of the dimension in sort order if there is no hierarchical relationship among dimension members. The top-level member name is generally the same as the dimension name if a hierarchical relationship exists.

trace allocations A feature of Profitability and Cost Management that enables you to visually follow the flow of financial data, either forwards or backwards, from a single intersection throughout the model.

trace level Defines the level of detail captured in the log file.

traceability The ability to track a metadata element to its physical source. For example, in Essbase Studio, a cube schema can be traced from its hierarchies and measure hierarchies, to its dimension elements, date/time elements, and measures, and ultimately, to its physical source elements.

traffic lighting Color-coding of report cells, or pins based on a comparison of two dimension members, or on fixed limits.

transformation (1) Transforms artifacts so that they function properly in the destination environment after application migration. (2) In data mining, modifies data (bidirectionally) flowing between the cells in the cube and the algorithm.

translation *See currency conversion*.

Transmission Control Protocol/Internet Protocol (TCP/IP) A standard set of communication protocols linking computers with different operating systems and internal architectures. TCP/IP utilities are used to exchange files, send mail, and store data to various computers that are connected to local and wide area networks.

transparent login Logs in authenticated users without launching the login screen.

transparent partition A shared partition that enables users to access and change data in a remote database as though it is part of a local database

triangulation A means of converting balances from one currency to another via a third common currency. In Europe, this is the euro for member countries. For example, to convert from French franc to Italian lira, the common currency is defined as European euro. Therefore, in order to convert balances from French franc to Italian lira, balances are converted from French franc to European euro and from European euro to Italian lira.

triggers An Essbase feature whereby data is monitored according to user-specified criteria which when met cause Essbase to alert the user or system administrator.

trusted password A password that enables users authenticated for one product to access other products without reentering their passwords.

trusted user Authenticated user.

tuple MDX syntax element that references a cell as an intersection of a member from each dimension. If a dimension is omitted, its top member is implied. Examples: (Jan); (Jan, Sales); ([Jan], [Sales], [Cola], [Texas], [Actual])

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

typed measure In Essbase, a member tagged as "Text" or "Date" in the dimension where measures are represented. The cell values are displayed as predefined text or dates.

unary operator A mathematical indicator (+, -, *, /, %) associated with an outline member. The unary operator defines how the member is calculated during a database roll-up.

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

Uniform Resource Locator The address of a resource on the Internet or an intranet.

unique member name A non-shared member name that exists only once in a database outline.

unique member outline A database outline that is not enabled for duplicate member names.

upgrade The process of replacing an earlier software release with a current release or replacing one product with another.

upper-level block A type of data block wherein at least one of the sparse members is a parent-level member.

user directory A centralized location for user and group information. Also known as a repository or provider.

user variable Dynamically renders data forms based on a user's member selection, displaying only the specified entity. For example, user variable named Department displays specific departments and employees.

user-defined attribute (UDA) User-defined attribute, associated with members of an outline to describe a characteristic of the members. Users can use UDAs to return lists of members that have the specified UDA associated with them.

user-defined member list A named, static set of members within a dimension defined by the user.

validation A process of checking a business rule, report script, or partition definition against the outline to make sure that the object being checked is valid. For example, in FDM, validation rules ensure that certain conditions are met after data is loaded from FDM to the target application.

value dimension Used to define input value, translated value, and consolidation detail.

variance Difference between two values (for example, planned and actual value).

varying attribute An attribute association that changes over one or more dimensions. It can be used to track a value in relation to these dimensions; for example, the varying attribute Sales Representative, associated with the Product dimension, can be used to track the value Customer Sales of several different sales representatives in relation to the Time dimension. Varying attributes can also be used for member selection, such as finding the Products that a Sales Representative was responsible for in May.

version Possible outcome used within the context of a scenario of data. For example, Budget - Best Case and Budget - Worst Case where Budget is scenario and Best Case and Worst Case are versions.

view Representation of either a year-to-date or periodic display of data.

visual cue A formatted style, such as a font or a color, that highlights specific types of data values. Data values may be dimension members; parent, child, or shared members; dynamic calculations; members containing a formula; read only data cells; read and write data cells; or linked objects.

Web server Software or hardware hosting intranet or Internet Web pages or Web applications.

weight Value assigned to an item on a scorecard that indicates the relative importance of that item in the calculation of the overall scorecard score. The weighting of all items on a scorecard accumulates to 100%. For example, to recognize the importance of developing new features for a product, the measure for New Features Coded on a developer's scorecard would be assigned a higher weighting than a measure for Number of Minor Defect Fixes.

wild card Character that represents any single character or group of characters (*) in a search string.

WITH section In MaxL DML, an optional section of the query used for creating re-usable logic to define sets or members. Sets or custom members can be defined once in the WITH section, and then referenced multiple times during a query.

work flow The steps required to process data from start to finish in FDM. The workflow consists of Import (loading data from the GL file), Validate (ensures all members are mapped to a valid account), Export (loads the mapped members to the target application), and Check (verifies accuracy of data by processing data with user-defined validation rules).

workbook An entire spreadsheet file with many worksheets.

Workspace Page A page created with content from multiple sources including documents, URL, and other content types. Enables a user to aggregate content from Oracle and non-Oracle sources.

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

ws.conf A configuration file for Windows platforms.

wsconf_platform A configuration file for UNIX platforms.

XML *See Extensible Markup Language (XML)*.

XOLAP An Essbase multidimensional database that stores only the outline metadata and retrieves all data from a relational database at query time. XOLAP supports aggregate storage databases and applications that contain duplicate member names.

Y axis scale Range of values on Y axis of charts displayed in Investigate Section. For example, use a unique Y axis scale for each chart, the same Y axis scale for all Detail charts, or the same Y axis scale for all charts in the column. Often, using a common Y axis improves your ability to compare charts at a glance.

Zero Administration Software tool that identifies version number of the most up-to-date plug-in on the server.

zoom Sets the magnification of a report. For example, magnify a report to fit whole page, page width, or percentage of magnification based on 100%.

ZoomChart Used to view detailed information by enlarging a chart. Enables you to see detailed numeric information on the metric that is displayed in the chart.

Index