

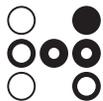
# IBM® DB2® OLAP Server™

Version

---



*OLAP Database Administrator's Guide  
Volume II*



Hyperion®

Hyperion Solutions Corporation

P/N: SC27-0789-00

© Copyright International Business Machines Corporation 1998, 2000. All rights reserved.

US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

© 1991–2000 Hyperion Solutions Corporation. All rights reserved.

U.S. Patent Number: 5,359,724

Hyperion, Essbase, and Arbor are registered trademarks, and Hyperion Solutions and Hyperion Essbase are trademarks of Hyperion Solutions Corporation.

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

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

Printed in the U.S.A.

# Contents

## Volume I

<b>Introduction</b> .....	xli
---------------------------	-----

## Part I: Designing Hyperion Essbase Applications

### Chapter 1: Introduction to Hyperion Essbase

About the Hyperion Essbase Product Family .....	1-1
Multidimensional Development Features .....	1-2
Architectural Features of Hyperion Essbase .....	1-3
Dynamic Dimensionality .....	1-3
Multithreaded Design and SMP .....	1-3
Multi-User Read and Write .....	1-3
Client-Server Overview .....	1-4
Hyperion Essbase Server .....	1-4
Hyperion Essbase Client .....	1-5

### Chapter 2: Steps for Implementing Hyperion Essbase

### Chapter 3: Multidimensional Concepts

Introducing OLAP .....	3-1
Introducing Dimensions and Members .....	3-3
Arranging Dimensions into Hierarchies .....	3-4

Defining Hyperion Essbase Terminology .....	3-5
Member Relationships, Generations, and Levels .....	3-5
Parents, Children, and Siblings.....	3-6
Descendants and Ancestors .....	3-6
Roots and Leaves .....	3-6
Generations and Levels.....	3-7
Identifying Values in a Multidimensional Database .....	3-8
Looking at Data from Different Perspectives .....	3-11
Designing and Creating a Simple Application .....	3-12
Organizing Multidimensional Data .....	3-13
Adding and Deleting Standard Dimensions and Stored Members .....	3-16
Adding Stored Members.....	3-16
Adding a Standard Dimension.....	3-18
Removing a Standard Dimension .....	3-19

## Chapter 4: Basic Architectural Elements

Attribute Dimensions and Standard Dimensions .....	4-1
Sparse and Dense Dimensions .....	4-2
Data Blocks and the Index System .....	4-4
Selection of Sparse and Dense Dimensions .....	4-8
Dense and Sparse Selection Scenarios .....	4-11
Scenario 1: A Database Consisting Entirely of Sparse Standard Dimensions .....	4-11
Scenario 2: A Database Consisting Entirely of Dense Standard Dimensions .....	4-12
Scenario 3: An Ideal Configuration of Dense and Sparse Standard Dimensions .....	4-13
Scenario 4: A Typical Multidimensional Problem .....	4-14
The Hyperion Essbase Solution .....	4-16

## Chapter 5: Designing a Single-Server Application

Designing an Application .....	5-2
Case Study: The Beverage Company .....	5-3
Planning and Analyzing .....	5-4
Identifying Source Data .....	5-4
Identifying User Requirements .....	5-5

Creating a Business Model .....	5-6
Identifying Analysis Objectives .....	5-6
Determining Dimensions and Members .....	5-7
Analyzing Database Design .....	5-10
Planning for Security in a Multi-User Environment .....	5-17
Drafting an Outline .....	5-17
Building an Outline.....	5-19
Defining Dimension and Member Properties .....	5-19
Designing an Outline to Optimize Performance.....	5-23
Loading Test Data .....	5-25
Defining and Testing Calculations .....	5-25
Consolidation Paths .....	5-26
Consolidation Ordering .....	5-27
Shared Members and Consolidation .....	5-28
Time and Accounts Calculations .....	5-28
Accounts Dimension Calculation .....	5-29
Time Balance Properties.....	5-29
Variance Reporting.....	5-31
Formulas .....	5-31
Dynamic Calculations .....	5-33
Two-Pass Calculations .....	5-34
Defining Reports .....	5-36
Verifying the Design .....	5-37

## **Chapter 6: Designing Partitioned Applications**

Introducing Hyperion Essbase Partitioning .....	6-2
Data Sources and Data Targets.....	6-3
Overlapping Partitions.....	6-5
What Is a Partition? .....	6-6
Workflow for Partitioning a Database.....	6-7
When to Partition a Database .....	6-8
When Not to Partition a Database .....	6-9
Determining Which Data to Partition .....	6-9

Deciding Which Type of Partition to Use .....	6-10
Replicated Partitions .....	6-10
Rules for Replicated Partitions .....	6-11
Advantages of Replicated Partitions .....	6-12
Disadvantages of Replicated Partitions .....	6-13
Performance Considerations for Replicated Partitions .....	6-13
When to Use a Replicated Partition .....	6-15
Replicated Partitions and Port Usage.....	6-15
Transparent Partitions .....	6-15
Rules for Transparent Partitions .....	6-16
Advantages of Transparent Partitions.....	6-18
Disadvantages of Transparent Partitions .....	6-18
Performance Considerations for Transparent Partitions .....	6-20
When to Use a Transparent Partition .....	6-20
Calculating Transparent Partitions.....	6-21
Transparent Partitions and Member Formulas.....	6-22
Transparent Partitions and Port Usage.....	6-23
Linked Partitions .....	6-23
Advantages of Linked Partitions.....	6-25
Disadvantages of Linked Partitions .....	6-26
When to Use a Linked Partition.....	6-26
Linked Partitions and Port Usage .....	6-26
Questions to Help You Choose a Partition Type .....	6-27
Using Attributes in Partitions .....	6-28
Setting Up Security for Partitioned Databases .....	6-30
Setting Up Security for Replicated Partitions .....	6-30
Setting Up Security for Transparent Partitions .....	6-31
Setting Up Security for Linked Partitions .....	6-32
Scenarios for Designing Partitioned Databases .....	6-33
Scenario 1: Partitioning an Existing Database .....	6-33
Scenario 2: Connecting Existing Related Databases .....	6-36
Scenario 3: Linking Two Databases .....	6-38

## Part II: Building Hyperion Essbase Applications

### Chapter 7: Creating Applications and Databases

Hyperion Essbase Applications .....	7-2
Multidimensional Database Outlines .....	7-3
Data Load and Dimension Build Rules .....	7-3
Calc Scripts .....	7-4
Report Scripts .....	7-4
Security Definitions .....	7-4
Process for Creating Databases .....	7-5
Starting and Stopping Applications and Databases .....	7-5
Using the Application Desktop Window .....	7-7
Deciding Where to Build an Application .....	7-9
Creating Applications and Databases .....	7-9
Creating a New Application .....	7-10
Creating a New Database .....	7-11
Rules for Naming Applications and Databases .....	7-12
Annotating a Database .....	7-13
Using Dynamic Calculations .....	7-14
Using Substitution Variables .....	7-14
Guidelines for Setting Substitution Variables .....	7-15
Setting a Substitution Variable .....	7-15
Deleting a Substitution Variable .....	7-17
Updating a Substitution Variable .....	7-18
Using Location Aliases .....	7-19
Creating Location Aliases .....	7-20
Editing or Deleting Location Aliases .....	7-22

### Chapter 8: Creating and Changing Database Outlines

Creating Outlines .....	8-2
Opening Outlines .....	8-2
Copying Outlines .....	8-4
Verifying and Saving Outlines .....	8-5
Verifying Outlines .....	8-7
Saving Outlines .....	8-7
Setting Dense and Sparse Data Storage .....	8-9

Renaming Dimensions and Members .....	8-10
Importing and Exporting 2.x Outlines .....	8-10
Exporting Outlines .....	8-11
Importing Outlines .....	8-13
Adding Dimensions and Members to Outlines .....	8-15
Rules for Naming Dimensions and Members .....	8-15
Adding Dimensions to Outlines .....	8-18
Adding Members to Dimensions .....	8-19
Positioning Dimensions and Members .....	8-20
Sorting Dimensions and Members .....	8-21
Moving Members .....	8-22
Naming Generations and Levels .....	8-23
Customizing the Outline Editor .....	8-24
Making Members Case-Sensitive .....	8-24
Customizing the View .....	8-25
Setting the Outline Font .....	8-26
Confirming Changes to Dimensions and Members .....	8-27

## Chapter 9: Setting Dimension and Member Properties

Setting the Dimension Type .....	9-2
Tagging a Time Dimension .....	9-2
Setting the Time Property .....	9-3
Tagging an Accounts Dimension .....	9-3
Setting the Accounts Property .....	9-4
Introducing Time Balance Properties .....	9-4
Introducing Skip Properties .....	9-6
Setting Time Balance Properties.....	9-8
Setting Variance Reporting Properties .....	9-9
Setting Hyperion Essbase Currency Conversion Properties.....	9-10
Tagging a Country Dimension .....	9-11
Tagging a Currency Partition .....	9-12
Tagging an Attribute Dimension .....	9-13
Setting the Attribute Property .....	9-13
Setting the Attribute Type.....	9-14
Setting Two-Pass Calculation Properties .....	9-15
Introducing Member Consolidation Properties .....	9-16

Setting Member Consolidation Properties .....	9-18
Setting Storage Properties .....	9-20
Storing Data .....	9-20
Dynamically Calculating Data .....	9-21
Creating Label Only Members .....	9-22
Sharing Members .....	9-23
Rules for Shared Members .....	9-23
Implied Sharing .....	9-23
Setting the Shared Member Property .....	9-25
Setting UDAs .....	9-25
Rules for UDAs .....	9-25
Creating UDAs .....	9-26
Setting Comments on Dimensions and Members .....	9-27
Setting Formulas for Dimensions and Members .....	9-28

## Chapter 10: Working with Attributes

About Attributes .....	10-2
About Attribute Dimensions .....	10-4
Members of Attribute Dimensions .....	10-5
Attribute Types .....	10-6
Differences Between Attribute and Standard Dimensions .....	10-7
Differences Between Attributes and UDAs .....	10-9
Introducing the Attribute Calculations Dimension .....	10-12
Attribute Design Considerations .....	10-12
Using Attribute Dimensions .....	10-12
Using Alternative Design Approaches .....	10-13
Outline Performance Considerations .....	10-14
Working with the Names of Members of Attribute Dimensions .....	10-15
Defining a Prefix or Suffix Format for the Names of Members of Attribute Dimensions .....	10-16
Changing Member Names of the Attribute Calculations Dimension .....	10-19
Changing the Member Names for Boolean Attribute Dimensions .....	10-21
Setting the Member Name Format of Date Attribute Dimensions .....	10-22
Assigning the Names of Members of Numeric Attribute Dimensions to Ranges of Values .....	10-23

Defining Attributes Manually .....	10-25
Associating Base Dimensions with Attribute Dimensions .....	10-26
Associating Base Dimension Members with Attributes .....	10-28
Calculating Attribute Data .....	10-30
Attribute Calculations Dimension .....	10-31
Default Attribute Calculations Members .....	10-32
An Attribute Calculation Example .....	10-34
Accessing Attribute Calculations Members Using the Spreadsheet .....	10-35
Calculation and Retrieval Performance Considerations .....	10-35
Using Attributes in Calculation Formulas .....	10-36

## Chapter 11: Creating and Managing Aliases

Introducing Aliases and Alias Tables .....	11-1
Rules for Aliases .....	11-2
Introducing Alias Tables .....	11-3
Format of an Alias Table .....	11-3
Creating Aliases for Members .....	11-4
Creating Aliases for Member Combinations .....	11-5
Creating New Alias Tables .....	11-7
Setting Alias Tables .....	11-7
Copying Existing Alias Tables .....	11-9
Renaming Alias Tables .....	11-9
Deleting and Clearing Alias Tables .....	11-10
Deleting Alias Tables .....	11-10
Clearing Alias Tables .....	11-11
Importing and Exporting Alias Tables .....	11-11
Importing an Alias Table .....	11-12
Exporting an Alias Table .....	11-13

## Chapter 12: Linking Objects to Hyperion Essbase Data

What Are Linked Reporting Objects? .....	12-1
How Do Linked Reporting Objects Work with Hyperion Essbase? .....	12-3
Managing Linked Reporting Objects .....	12-4
Providing Access to Linked Reporting Objects .....	12-4
Limiting LRO File Sizes .....	12-5
Viewing and Deleting Linked Reporting Objects .....	12-6

## Chapter 13: Introducing Dynamic Dimension Building

Workflow for Creating Dimensions Using Rules Files .....	13-2
Introduction to Build Methods .....	13-3
Using Generation References .....	13-5
Null Processing with Generation References .....	13-7
Using Level References .....	13-8
Null Processing with Level References .....	13-10
Using Parent/Child References .....	13-11
Adding a List of New Members .....	13-12
Adding Members Based Upon String Matches .....	13-13
Adding Members as Siblings of the Lowest Level .....	13-15
Adding All New Members to a Specified Parent .....	13-16
Building Attribute Dimensions and Associating Attributes .....	13-18
Building Attribute Dimensions .....	13-19
Associating Attributes .....	13-19
Updating Attribute Associations .....	13-21
Working with Multilevel Attribute Dimensions .....	13-22
Working With Numeric Ranges .....	13-25
Building the Attribute Dimension .....	13-26
Associating the Base Dimension Members with Their Range Attributes .....	13-27
Working With Ranges .....	13-29
Rules Summary for Building Attribute and Base Dimensions .....	13-30
Building Shared Members Using a Rules File .....	13-32
Sharing Members at the Same Generation .....	13-33
Creating Shared Members Using Generation References .....	13-33
Creating Shared Members Using Level References .....	13-35
Creating Shared Members Using Parent/Child References .....	13-36
Sharing Members at Different Generations .....	13-37
Creating Shared Members Using Level References .....	13-38
Creating Shared Members Using Parent/Child References .....	13-39
Sharing Members with Branches .....	13-40
Creating Shared Members Using Level References .....	13-40
Creating Shared Members Using Parent/Child References .....	13-42

Building Multiple Roll-Ups Using Level References .....	13-43
Creating Shared Roll-Ups from Multiple Data Sources	
Using Parent/Child References .....	13-44
Security and Multi-User Considerations .....	13-46

## **Chapter 14: Building Dimensions Using a Rules File**

About Dimensions and Rules Files .....	14-2
Step 1: Defining Dimensions .....	14-2
Naming New Dimensions .....	14-3
Naming a New Standard Dimension .....	14-4
Naming New Attribute Dimensions .....	14-4
Setting Dimension Properties .....	14-6
Step 2: Choosing the Build Method .....	14-9
Step 3: Specifying Changes to Dimensions .....	14-10
Step 4: Setting Rules File Field Types .....	14-12
Setting Field Information .....	14-13
Step 5: Setting Global Options .....	14-20
Step 6: Validating Dimension Build Rules .....	14-21
Manipulating the Data Source .....	14-23
Using Dynamic References .....	14-23
Setting Member Properties .....	14-24
Performing Dimension Builds .....	14-26
Updating Dimensions in Outline Editor .....	14-26
Building Dimensions Without Connecting to the Server .....	14-29
Debugging Dimension Builds .....	14-29
How Hyperion Essbase Builds Dimensions .....	14-31

## **Chapter 15: Estimating Disk and Memory Requirements for a Database**

Understanding How Hyperion Essbase Stores Data .....	15-1
Determining Disk Space Requirements .....	15-3
General Disk Space Requirements for a Database .....	15-3
Determining the Expanded Block Size of Stored Data.....	15-5
Estimating the Size of Compressed Data Blocks.....	15-5
Estimating the Number of Data Blocks .....	15-6
Estimating the Size of the Compressed Data Files .....	15-7

Estimating the Size of the Index.....	15-8
Estimating Fixed-Size Overhead.....	15-8
Calculating a Fragmentation Allowance .....	15-10
Calculating a Data Recovery Area .....	15-10
Estimating the Size of the Outline.....	15-10
Allowing for Restructuring and Migration Work Areas .....	15-11
Estimating Disk Space for a Database.....	15-12
Considerations When Using Partitioning .....	15-12
Considerations When Using Linked Reporting Objects .....	15-13
Total Disk Requirement .....	15-13
Determining Memory Requirements.....	15-14
General Memory Requirements Per Database .....	15-14
Sizing Your Index, Data File, and Data Caches .....	15-15
Using Cache Memory Locking .....	15-16
Estimating Cache Sizes .....	15-16
Fine Tuning Your Cache Settings .....	15-18
Sizing Cache-Related Overhead.....	15-20
Estimating Additional Memory Requirements for Database Operations .....	15-20
Estimating Additional Memory Requirements for Data Retrievals .....	15-21
Estimating Additional Memory Requirements for Calculations .....	15-25
Estimating Total Memory Requirements .....	15-26

## **Chapter 16: Building and Maintaining Partitions**

Opening the Partition Manager .....	16-2
Creating a New Partition .....	16-3
Creating a Replicated Partition .....	16-4
Specifying the Partition Type and Connection Information.....	16-5
Setting the User Name and Password.....	16-6
Defining the Areas in a Partition .....	16-8
Manually Defining an Area .....	16-9
Entering an Area Using the Member Selection Dialog Box .....	16-10
Mapping Data Source Members to Data Target Members .....	16-11
Mapping Members with Different Names.....	16-12
Mapping Data Cubes with Extra Dimensions .....	16-12
Manually Defining Mappings.....	16-17

Using the Member Selection Dialog Box to Enter Mappings .....	16-17
Importing Member Mappings .....	16-19
Creating a Transparent Partition .....	16-20
Specifying the Partition Type and Connection Information .....	16-21
Setting the User Name and Password .....	16-21
Defining the Areas in a Transparent Partition .....	16-22
Manually Defining an Area .....	16-23
Using the Member Selection Dialog Box to Enter an Area.....	16-23
Mapping Data Source Members to Data Target Members .....	16-25
Manually Defining the Mapping.....	16-27
Using the Member Selection Dialog Box to Enter Mappings .....	16-27
Mapping Attributes Associated with Members .....	16-28
Creating a Linked Partition .....	16-31
Specifying the Partition Type and Connection Information .....	16-31
Setting the User Name and Password .....	16-32
Defining the Areas in a Partition .....	16-33
Manually Defining an Area .....	16-34
Using the Member Selection Dialog Box to Enter an Area.....	16-34
Mapping Data Source Members to Data Target Members .....	16-35
Manually Defining Mappings.....	16-37
Using the Member Selection Dialog Box to Enter Mappings .....	16-37
Validating the Partition .....	16-38
Saving the Partition Definition .....	16-40
Testing Partitions .....	16-41
Creating Advanced Area-Specific Mappings (Optional) .....	16-42
Example 1: Advanced Area-Specific Mapping .....	16-42
Example 2: Advanced Area-Specific Mapping .....	16-44
Specifying Area-Specific Mapping.....	16-46
Introducing Outline Synchronization .....	16-47
Source Outline and Target Outline .....	16-47
How Hyperion Essbase Tracks Changes .....	16-48
Synchronizing Outlines .....	16-50
How Shared Members are Treated in Outline Synchronization .....	16-53
Editing Partitions .....	16-57
Introducing the Updating of Replicated Partitions .....	16-58

Updating Replicated Partitions .....	16-60
Updating Replicated Partitions Using ESSCMD .....	16-62
Troubleshooting Partitions .....	16-62

## Part III: Designing and Building a Security System

### Chapter 17: Managing Security at Global and User Levels

Multi-Layered Security and Privileges .....	17-1
The ESSBASE.SEC Security File .....	17-2
Privileges Available at Global and User Levels .....	17-2
Managing Security at the User and Group Layer .....	17-4
User and Group Types .....	17-5
Managing Users and Groups .....	17-7
Creating, Editing, and Copying Users and Groups .....	17-9
Creating a New User .....	17-9
Editing a User .....	17-12
What are “Groups” and How Do You Create Them? .....	17-14
Creating or Editing a Group .....	17-14
Copying an Existing Security Profile .....	17-16
Copying a User or Group Profile.....	17-16
Copying a Group Profile .....	17-18
Deleting Users and Groups .....	17-19
Renaming Users and Groups .....	17-21
Modifying User Application and Database Access Settings .....	17-22
Assigning Database Access to a User.....	17-24
Viewing and Modifying User Access Privileges .....	17-27
Viewing and Modifying User Application Access.....	17-27
Viewing and Modifying User Database Access .....	17-28
Managing Security at the Global Access Layer .....	17-29
Defining Global Application Access .....	17-31
Minimum Database Access .....	17-32
Allow Settings .....	17-33
Defining Global Database Access .....	17-34
Assigning Global Access Settings Per User .....	17-35
Application Designer Privilege .....	17-35
Database Designer Privilege.....	17-36

Managing User Activity at the Server Level .....	17-36
Managing Locks .....	17-37
Disconnecting Users .....	17-38
Managing Passwords and User Names .....	17-39
Viewing or Activating Disabled User Names .....	17-41

## Chapter 18: Controlling Access to Database Cells

Privileges at the Database Filter Layer .....	18-2
Defining Filters .....	18-3
Creating a New Filter .....	18-3
Mini-Tutorial .....	18-6
Filtering Whole Members vs. Filtering Member Combinations.....	18-8
Filtering with Attribute Functions .....	18-10
Editing a Filter .....	18-11
Copying a Filter .....	18-12
Renaming a Filter .....	18-13
Deleting a Filter .....	18-14
Assigning Filters.....	18-15
Overlapping Filter Definitions .....	18-16
Overlapping Access Definitions .....	18-17

## Chapter 19: Security Examples

Security Problem 1 .....	19-1
Security Problem 2 .....	19-2
Security Problem 3 .....	19-3
Security Problem 4 .....	19-4
Security Problem 5 .....	19-5

## Part IV: Loading Data

### Chapter 20: Introducing Data Loading

Introduction to Data Sources .....	20-1
How Does Hyperion Essbase Read a Data Source? .....	20-2
Valid Data Fields .....	20-3
Valid Member Fields .....	20-5
Invalid Member or Data Fields .....	20-6

Setting File Delimiters .....	20-6
Ignored Characters .....	20-7
Introduction to Rules Files .....	20-8
When to Use Data Load Rules .....	20-8
How to Create Data Load Rules .....	20-9
How Does Hyperion Essbase Execute Operations in a Rules File? .....	20-10
Rules for Rules File Data Sources .....	20-12
Rules for Dimension Fields .....	20-12
Rules for Member Fields .....	20-14
Quoting Member Names .....	20-14
Unknown Member Names .....	20-15
Rules for Data Fields .....	20-15
Assigning All Members .....	20-16
Empty Data Fields .....	20-16
Rules for Extra Fields .....	20-17
No Blank Fields .....	20-17
Each Record Must Have the Same Number of Fields .....	20-17
Rules for File Delimiters .....	20-18
Rules for Free-Form Data Sources .....	20-18
Formatting Ranges of Member Fields .....	20-19
Ranges Set Automatically .....	20-20
Data Values Out of Range .....	20-20
Duplicate Members in a Range .....	20-21
How Hyperion Essbase Reads Multiple Ranges .....	20-22
Formatting Columns .....	20-22
Symmetric Columns .....	20-22
Asymmetric Columns .....	20-23
Security and Multi-User Considerations .....	20-24

## Chapter 21: Setting up a Rules File to Manipulate Records

Selecting the Data Source .....	21-1
Connecting to the Server .....	21-2
Viewing the Application and Database .....	21-2
Opening the Data Prep Editor .....	21-3
Viewing Data Load Fields or Dimension Build Fields .....	21-4
Customizing the Data Prep Editor .....	21-4

Opening a Data Source .....	21-5
Opening a Text File .....	21-5
Opening a Spreadsheet File .....	21-7
Opening an SQL Data Source.....	21-8
Setting File Delimiters .....	21-10
Using Header Information .....	21-11
Defining Header Information in the Rules File .....	21-11
Defining Header Information in the Data Source for a Data Load .....	21-13
Selecting Records .....	21-15
Rejecting Records.....	21-16
Defining Multiple Select and Reject Criteria .....	21-18
Setting the Records Displayed .....	21-20
Validating and Saving Data Load Rules .....	21-21
Associating a Rules File with an Outline .....	21-22
Validating a Rules File .....	21-23
Saving Rules Files .....	21-24
Saving Under a Different Name .....	21-25

## **Chapter 22: Manipulating Fields Using a Rules File**

Selecting Multiple Fields .....	22-2
Ignoring Fields .....	22-2
Ignoring All Fields in a Column .....	22-3
Ignoring Fields Based on String Matching .....	22-4
Ordering Fields .....	22-5
Moving Fields .....	22-5
Joining Fields .....	22-6
Creating a New Field While Leaving Existing Fields Intact .....	22-7
Creating a New Field By Joining Fields .....	22-7
Copying Fields .....	22-8
Splitting Fields .....	22-9
Creating Additional Text Fields .....	22-10
Undoing Field Ordering .....	22-11

Mapping Fields to Member Names .....	22-12
Naming Fields .....	22-12
Replacing Text Strings .....	22-14
Replacing an Empty Field with Text .....	22-15
Changing the Case of Fields .....	22-16
Dropping Leading/Trailing White Space .....	22-18
Converting Spaces to Underscores .....	22-19
Adding Prefixes or Suffixes to Field Values .....	22-20
Defining a Column as a Data Field .....	22-21
Changing Data Values .....	22-22
Adding to and Subtracting from Existing Values .....	22-22
Clearing Existing Data Values .....	22-23
Scaling Data Values .....	22-25
Flipping Field Signs .....	22-27

## Chapter 23: Performing a Data Load

Prerequisites for Loading Data and Building Dimensions .....	23-2
Choosing the Data Sources Using the Hyperion Essbase Application Manager .....	23-3
Choosing SQL Data Sources .....	23-4
Choosing Text or Spreadsheet Files .....	23-5
Choosing the Data Sources Using Windows .....	23-7
Specifying How to Load Data or Build Dimensions .....	23-8
Using a Rules File with the Data Source .....	23-9
Building Dimensions Dynamically by Modifying the Outline .....	23-10
Updating the Database Outline in Batch Mode .....	23-12
Setting the Error Log File .....	23-12
Starting the Data Load or Dimension Build .....	23-12
Finishing the Data Load or Dimension Build .....	23-13
Complete Load .....	23-13
Partial Load .....	23-14
No Load .....	23-16
Tips for Loading Data .....	23-17
Where to Load Data .....	23-17
Loading a Range of Records .....	23-17
Loading Data Using a Spreadsheet .....	23-18

## Chapter 24: Debugging and Optimizing Data Loads

Debugging a Data Load .....	24-1
Not All Errors Are in the Log File .....	24-2
Data Loaded Incorrectly .....	24-2
Verifying that the Server Is Available .....	24-4
Verifying that the Data Source Is Available .....	24-4
Loading the Error Log File .....	24-5
Recovering from a Server Crash .....	24-5
Problems Validating a Rules File .....	24-6
Ignoring End of File Markers .....	24-6
Optimizing Data Loads .....	24-7
How Does Hyperion Essbase Load Data? .....	24-7
Grouping Sparse Member Combinations Together .....	24-8
Positioning Data in the Same Order as the Outline .....	24-10
Loading from the Server .....	24-11
Making the Data Source as Small as Possible .....	24-11
Making the Fields as Small as Possible .....	24-12

## Part V: Calculating Data

### Chapter 25: Introduction to Database Calculations

Understanding Database Calculations .....	25-1
Outline Calculation .....	25-2
Calc Script Calculation .....	25-3
Calculating a Multidimensional Database .....	25-4
Setting the Default Calculation .....	25-7
Calculating a Database .....	25-8
Running a Calculation .....	25-9
Canceling a Calculation .....	25-11
Considering Security .....	25-12

## Chapter 26: Developing Formulas

Using Formulas .....	26-1
Operators .....	26-3
Functions .....	26-4
Calculating Formulas .....	26-5
Guidelines for Formula Syntax .....	26-6
Creating Formulas .....	26-8
Building Formulas in Formula Editor .....	26-11
Opening Formula Editor .....	26-12
Displaying a Formula .....	26-13
Adding a Formula .....	26-13
Changing a Formula .....	26-13
Saving a Formula .....	26-14
Printing a Formula .....	26-14
Deleting a Formula .....	26-14
Inserting Text and Operators in a Formula .....	26-15
Inserting Members in a Formula .....	26-20
Searching for Members .....	26-22
Checking Syntax .....	26-25
Writing Formulas .....	26-27
Writing Basic Equations .....	26-28
Specifying Conditions .....	26-29
Examples of Specifying Conditions .....	26-31
Using Interdependent Values .....	26-33
Specifying a Member List or Range .....	26-35
Generating Member Lists .....	26-36
Performing Mathematical Operations .....	26-38
Calculating a Variance or Percentage Variance Between Actual and Budget Values .....	26-39
Calculating Statistics .....	26-41
Allocating and Forecasting Values .....	26-42
Using Range Functions .....	26-43
Using the Current Member Combination to Look Up Values .....	26-44
Calculating Financial Functions .....	26-45

Using Date and Time Functions .....	26-46
Using the Cross-Dimensional Operator (->).....	26-46
Using Substitution Variables .....	26-48
Working with Formulas in Partitions .....	26-49

## Chapter 27: Examples of Formulas

Calculating Period-to-Date Values .....	27-1
Calculating Rolling Values .....	27-3
Calculating Monthly Asset Movements .....	27-4
Testing for #MISSING Values .....	27-5
Calculating an Attribute Formula .....	27-6

## Chapter 28: Defining the Calculation Order

Data Storage in Data Blocks .....	28-1
Member Calculation Order .....	28-3
Member Relationships .....	28-4
Member Consolidation .....	28-5
Ordering Dimensions in the Database Outline .....	28-6
Placing Formulas on Members in the Database Outline.....	28-6
Using the Unary Operators *, /, and % .....	28-7
Avoiding Forward Calculation References .....	28-8
Block Calculation Order .....	28-11
Data Block Renumbering.....	28-14
Cell Calculation Order .....	28-14
Cell Calculation Order Examples .....	28-15
Cell Calculation Order for Formulas on a Dense Dimension .....	28-22
Calculation Passes .....	28-23
Calculating Shared Members .....	28-26

## Chapter 29: Dynamically Calculating Data Values

Understanding Dynamic Calculations .....	29-2
Understanding Dynamic Calc Members .....	29-2
Understanding Dynamic Calc And Store Members .....	29-3
Recalculation of Data.....	29-3
Effect of Updated Values on Recalculation.....	29-4
Retrieving the Parent Value of Dynamically Calculated Child Values .....	29-4

Benefitting from Dynamic Calculations .....	29-5
Using Dynamic Calculations .....	29-5
Choosing Which Values to Calculate Dynamically .....	29-6
Tagging Dense Dimension Members .....	29-7
Tagging Sparse Dimension Members .....	29-7
Tagging Two-Pass Members .....	29-8
Calc Scripts and Dynamic Calculations .....	29-8
Formulas and Dynamically Calculated Members .....	29-8
Dynamically Calculated Children of Regular Members .....	29-9
Reducing the Impact on Retrieval Time .....	29-10
Displaying a Retrieval Factor .....	29-10
Displaying a Summary of Dynamically Calculated Members .....	29-11
Increasing the Retrieval Buffer Size .....	29-12
Choosing Between Dynamic Calc and Dynamic Calc And Store .....	29-13
Recommendations for Sparse Dimension Members .....	29-13
Recommendations for Members with Specific Characteristics .....	29-14
Recommendations for Dense Dimension Members .....	29-15
Recommendations for Data with Many Concurrent Users .....	29-15
Considering the Effects of Dynamic Calculations .....	29-15
Calculation Order for Dynamic Calculations .....	29-16
Calculation Order for Dynamically Calculating Two-Pass Members .....	29-17
Calculation Order for Asymmetric Data .....	29-18
Using Dynamic Calculations with Standard Procedures .....	29-20
Clearing Data and Data Blocks .....	29-20
Copying Data .....	29-20
Converting Currencies .....	29-20
Loading Data .....	29-21
Exporting Data .....	29-21
Reporting Data .....	29-21
Including Dynamic Calc and Dynamic Calc And Store Members in Calc Scripts .....	29-21
Creating Dynamic Calc and Dynamic Calc And Store Members .....	29-22
Building Dimensions with Dynamic Calc Members .....	29-23
Restructuring a Database .....	29-23
Dynamically Calculating Data in Partitions .....	29-25

## Chapter 30: Calculating Time Series Data

Calculating First, Last, or Average Values .....	30-1
Specifying Accounts and Time Dimensions .....	30-2
Reporting the Last Value for Each Time Period .....	30-3
Reporting the First Value for Each Time Period .....	30-4
Reporting the Average Value for Each Time Period .....	30-4
Skipping #MISSING and Zero Values .....	30-5
Considering the Effects of First, Last, and Average Tags .....	30-6
Placing Formulas on Time and Accounts Dimensions .....	30-6
Calculating Period-to-Date Values .....	30-7
Using Dynamic Time Series Members .....	30-7
Enabling Dynamic Time Series Members .....	30-9
Disabling Dynamic Time Series Members .....	30-11
Specifying Alias Names for Dynamic Time Series Members .....	30-13
Applying Predefined Generation Names to Dynamic Time Series Members ....	30-14
Retrieving Period-to-Date Values .....	30-14
Using Dynamic Time Series Members in Partitions .....	30-16

## Chapter 31: Developing Calc Scripts

Using a Calc Script .....	31-2
Creating a Calc Script .....	31-3
Calculating Sample Basic Data .....	31-6
Running a Calc Script .....	31-7
Checking a Calculation .....	31-9
Building a Calc Script in Calc Script Editor .....	31-10
Implementing Outline Calculations .....	31-11
Controlling the Flow of Calculations .....	31-12
Declaring Data Variables .....	31-12
Specifying Global Settings for a Database Calculation .....	31-13
Adding Comments .....	31-15
Composing Calc Script Syntax .....	31-15
Opening Calc Script Editor .....	31-18
Adding a Calc Script .....	31-19
Changing a Calc Script .....	31-20
Saving a Calc Script .....	31-24

Running a Calc Script.....	31-31
Printing a Calc Script .....	31-35
Deleting a Calc Script.....	31-35
Using Formulas in a Calc Script.....	31-36
Basic Equations .....	31-37
Conditional Equations .....	31-37
Interdependent Formulas .....	31-38
Inserting Text and Operators in a Calc Script .....	31-39
Associating a Calc Script with a Database .....	31-43
Checking Syntax .....	31-48
Using a Calc Script to Control Intelligent Calculation .....	31-50
Grouping Formulas and Calculations.....	31-50
Calculating a Series of Member Formulas .....	31-51
Calculating a Series of Dimensions .....	31-51
Using Substitution Variables.....	31-52
Clearing Data .....	31-53
Copying Data .....	31-54
Calculating a Subset of a Database .....	31-54
Calculating Lists of Members .....	31-55
Using the FIX Command .....	31-55
Writing Calc Scripts for Partitions .....	31-57
Calculating Multiple Databases.....	31-58

## Chapter 32: Examples of Calc Scripts

Calculating Variance .....	32-2
Calculating a Subset of a Database .....	32-3
Loading New Budget Values .....	32-5
Calculating Product and Market Share Values .....	32-6
Allocating Costs Across Products .....	32-7
Allocating Values Within or Across Dimensions .....	32-9
Allocating Within a Dimension .....	32-9
Allocating Across Multiple Dimensions .....	32-13
Goal Seeking Using the LOOP Command.....	32-17
Forecasting Future Values.....	32-21

## Chapter 33: Optimizing Calculations

Designing for Calculation Performance .....	33-2
Block Size and Block Density .....	33-2
Order of Sparse Dimensions .....	33-3
Incremental Data Loading Considerations .....	33-3
Performance for Database Outlines with Two or More Flat Dimensions .....	33-4
Monitoring and Tracing Calculations .....	33-4
SET MSG SUMMARY and SET MSG DETAIL .....	33-4
SET NOTICE .....	33-5
Using Formulas .....	33-5
Optimizing Formulas on Sparse Dimensions in Large Database Outlines .....	33-8
Assigning Constants to Members in a Sparse Dimension .....	33-8
Using a Cross-Dimensional Operator (->) .....	33-10
On the Left Side of an Equation .....	33-10
In Equations in a Dense Dimension .....	33-11
Understanding Bottom-Up Versus Top-Down Calculation .....	33-12
Implementing Calc Script Techniques .....	33-13
Setting Memory Cache Sizes .....	33-14
Using the Calculator Cache .....	33-15
Understanding Calculator Cache Options .....	33-16
Calculating the Required Size of the Calculator Cache .....	33-18
Calculating the Database for the First Time .....	33-21
Using the Calculator Cache for Large, Flat Database Outlines .....	33-22
Locking Blocks During Calculation .....	33-22
Considering Multiple Users .....	33-23
Using Two-Pass Calculation .....	33-24
Using Two-Pass on a Default Calculation .....	33-27
Scenario A .....	33-29
Scenario B .....	33-30
Using a Calc Script for Two-Pass Calculations .....	33-31
Option 1: Using Intelligent Calculation with a Large Index .....	33-32
Option 2: Using Intelligent Calculation with a Small Index .....	33-33
Option 3: Not Using Intelligent Calculation .....	33-34

Calculating #MISSING Values .....	33-35
Loading Data at Parent Levels .....	33-38

## **Chapter 34: Using Intelligent Calculation to Optimize Calculation**

Introducing Intelligent Calculation .....	34-1
Benefits of Intelligent Calculation .....	34-2
Intelligent Calculation and Data Block Status .....	34-2
Maintaining Clean and Dirty Status .....	34-4
Limitations of Intelligent Calculation .....	34-4
Using Intelligent Calculation .....	34-5
Turning Intelligent Calculation On and Off .....	34-5
Using Intelligent Calculation for a Default, Full Calculation .....	34-5
Calculating for the First Time .....	34-6
Recalculating .....	34-6
Using Intelligent Calculation for a Partial, Calc Script Calculation .....	34-6
Using the SET CLEARUPDATESTATUS Command .....	34-7
Considerations for Using SET CLEARUPDATESTATUS .....	34-8
Examples Using SET CLEARUPDATESTATUS .....	34-9
Calculating Data Blocks .....	34-10
Calculating a Dense Dimension .....	34-10
Calculating a Sparse Dimension .....	34-11
Handling Concurrent Calculations .....	34-12
Handling Multiple-Pass Calculations .....	34-13
Considering the Effects of Intelligent Calculation .....	34-18
Changing a Formula or Accounts Property .....	34-18
Using Relationship and Financial Functions .....	34-19
Restructuring a Database .....	34-19
Copying and Clearing Data .....	34-19
Converting Currencies .....	34-20

## Volume II

### Part VI: Reporting on Data

#### Chapter 35: Quick Start to Report Scripts

Creating a Simple Report Script .....	35-2
Parts of Report Scripts and Reports .....	35-5
How the Report Extractor Retrieves Data .....	35-6
Parts of a Report .....	35-7
Parts of a Report Script .....	35-9
Planning a Report .....	35-10
Security and Multi-User Issues .....	35-10
Creating and Editing Report Scripts .....	35-11
Creating New Report Scripts .....	35-12
Saving Report Scripts .....	35-13
Opening Report Scripts from the Application Manager .....	35-14
Opening Report Scripts from an Application or Database Directory .....	35-16
Finding Text .....	35-17
Replacing Text .....	35-18
Cutting, Copying, and Pasting Text .....	35-20
Deleting Text .....	35-20
Running Report Scripts .....	35-21
Choosing the Report Output .....	35-21
Sending the Report Output to a Window .....	35-21
Sending Report Output to a Printer .....	35-22
Sending Report Output to a File .....	35-24
Choosing Report Databases .....	35-26
Running the Report .....	35-27
Running Multiple Report Script Files .....	35-27
Executing a Report String Without Creating a Report Script .....	35-27
Developing Free-Form Reports .....	35-28

## Chapter 36: Developing Report Scripts

Syntax Guidelines .....	36-2
Designing the Page Layout .....	36-3
Creating Page, Column, and Row Headings .....	36-4
Modifying Headings.....	36-6
Creating Symmetric and Asymmetric Reports .....	36-7
Overriding Default Column Groupings.....	36-8
Changing Column Headings.....	36-8
Formatting .....	36-8
Formatting Report Pages .....	36-9
Setting the Page Length, Width, and Centering .....	36-9
Inserting Page Breaks .....	36-10
Formatting Page, Column, and Row Headings .....	36-10
Specifying Column Formats.....	36-10
Accommodating Long Column and Row Names.....	36-13
Suppressing Page, Column, and Row Formatting.....	36-14
Repeating Row Names .....	36-15
Formatting Reports with Tab Delimiters.....	36-15
Adding Totals and Subtotals .....	36-16
Totaling Columns .....	36-16
Numbering Columns .....	36-18
Totaling Rows .....	36-20
Changing How Data is Displayed .....	36-22
Underlining.....	36-22
Suppressing Data Formatting .....	36-23
Indenting.....	36-24
Inserting Custom Titles .....	36-24
Replacing Missing Text or Zeros with Labels .....	36-25
Adding Blank Spaces .....	36-26
Changing How Data Values Display.....	36-26
Selecting and Sorting Members .....	36-27
Selecting Members .....	36-27
Selecting Members Using Generation and Level Names.....	36-29
Selecting Dynamic Time Series Members .....	36-31
Selecting Members Using Boolean Operators .....	36-32
Selecting Members Using Substitution Variables .....	36-33

Selecting Members Using Attributes .....	36-35
Selecting Members Associated with a Specific Attribute .....	36-36
Selecting Members by a Specific Date .....	36-36
Selecting Members Using User-Defined Attributes .....	36-37
Selecting Members Using Wildcards .....	36-38
Selecting Members Using Static Member Names .....	36-39
Suppressing Shared Members .....	36-41
Selecting Alias Names for Members .....	36-42
Sorting Members .....	36-44
Restricting and Ordering Data Values .....	36-45
Defining the Order of Operation .....	36-46
Using TOP, BOTTOM, and ORDERBY with Sorting Commands .....	36-46
Restricting Data Ranges .....	36-46
Ordering Data .....	36-47
Using ORDERBY with Formatting Commands.....	36-48
Specifying Rows to Return .....	36-48
How Other Report Configurations Affect Row Specifications .....	36-50
Using TOP and BOTTOM with Formatting Commands.....	36-51
Converting Data to a Different Currency .....	36-51
Generating Reports Using the C, Visual Basic, and Grid APIs .....	36-52

## Chapter 37: Examples of Report Scripts

Sample 1: Creating a Different Format for Each Page .....	37-2
Sample 2: Handling Missing Values .....	37-4
Sample 3: Nesting Columns .....	37-6
Sample 4: Grouping Rows .....	37-8
Sample 5: Reporting on Different Combinations of Data .....	37-14
Sample 6: Formatting Different Combinations of Data .....	37-17
Sample 7: Using Aliases .....	37-20
Sample 8: Creating Custom Headings and % Characters .....	37-23
Sample 9: Creating Custom Page Headings .....	37-27
Sample 10: Using Formulas .....	37-32
Sample 11: Placing Two Page Layouts on the Same Page .....	37-35
Sample 12: Formatting for Data Export .....	37-37
Sample 13: Creating Asymmetric Columns .....	37-39

Sample 14: Calculating Columns .....	37-40
Sample 14-A: Basic Calculated Columns .....	37-41
Sample 14-B: Asymmetric Columns .....	37-42
Sample 15: Calculating Rows .....	37-44
Sample 15-A: Basic Calculated Row .....	37-44
Sample 15-B: Calculated Rows and Missing Relationships .....	37-45
Sample 15-C: Averaging Rows .....	37-47
Sample 16: Sorting by Top or Bottom Data Values .....	37-51
Sample 16-A: Bottom Data Values .....	37-51
Sample 16-B: Top Data Values .....	37-53
Sample 17: Restricting Rows .....	37-54
Sample 18: Ordering Data Values .....	37-55
Sample 19: Narrowing Member Selection Criteria .....	37-57
Sample 20: Using Attributes in Member Selection .....	37-58
Sample 21: Using the WITHATTR Command in Member Selection .....	37-60

## **Chapter 38: Optimizing Your Reports**

Setting Configurable Variables .....	38-2
Setting the Retrieval Buffer Size .....	38-2
Setting the Retrieval Sort Buffer Size .....	38-4
Setting the NumericPrecision Variable .....	38-5
Generating Symmetric Reports .....	38-6
Organizing Members to Optimize Data Extraction .....	38-7
Working with Database Outlines .....	38-8
Accessing Outlines that Contain Dynamic or Transparent Members .....	38-8
Ensuring that Members in the Report Script Synchronize with the Database Outline .....	38-8

## **Chapter 39: Copying Data Subsets and Exporting Data to Other Programs**

Copying a Database Subset to Personal Essbase .....	39-1
Summary of Steps .....	39-2
Creating a New Application and Database .....	39-3
Copying the Outline File from Your Source Database .....	39-4
Copying the Outline File Using Application Manager .....	39-5
Copying the Outline File Using the Operating System .....	39-6

Creating an Output File Containing the Required Data Subset ..... 39-7  
Loading the Output File Into the New Database ..... 39-11  
Using Report Scripts for Data Exporting ..... 39-13  
Importing Data Into Other Databases ..... 39-16

## **Part VII: Managing Multidimensional Hyperion Essbase Data Storage**

### **Chapter 40: Introducing the Hyperion Essbase Kernel**

About the Hyperion Essbase Kernel ..... 40-1  
    Index Manager ..... 40-3  
        Allocation Manager ..... 40-4  
    Data Block Manager ..... 40-5  
    LRO Manager ..... 40-5  
    Lock Manager ..... 40-6  
    Transaction Manager ..... 40-6  
    Hyperion Essbase Kernel Startup ..... 40-7  
    Fatal Error Handling ..... 40-7  
Data Compression ..... 40-8  
    Using Bitmap Data Compression ..... 40-8  
    Using RLE Data Compression ..... 40-9  
    Deciding Which Type of Compression to Use ..... 40-10  
    Deciding When to Disable Data Compression ..... 40-10  
    Changing Data Compression Settings ..... 40-10  
    Checking the Compression Ratio ..... 40-11  
    Data Compression ..... 40-11  
Storage Allocation ..... 40-11  
    Fragmentation ..... 40-14  
    Index and Data File Sizes ..... 40-14  
    Database Restructuring ..... 40-14  
    The Hyperion Essbase Process for Restructuring Data ..... 40-16  
    Optimization of Restructure Operations ..... 40-18  
        Actions That Improve Performance ..... 40-18  
        Incremental Restructuring ..... 40-19

Restructuring Considerations .....	40-20
Outline Change Log .....	40-20
IBM Relational Storage Manager.....	40-20
Hyperion Essbase Partitioning Option .....	40-21
Impact of Common Outline Changes .....	40-21
The Hyperion Essbase Process for Restructuring After Outline Changes .....	40-27
Saving a Modified Outline .....	40-27
Saving an Outline with One or More New Standard Dimensions.....	40-28
Saving an Outline with One or More Deleted Standard Dimensions.....	40-29

## **Chapter 41: Specifying Hyperion Essbase Kernel Settings**

About Hyperion Essbase Kernel Settings .....	41-2
Specifying and Changing Hyperion Essbase Kernel Settings .....	41-3
ESSCMDs That Control Database Settings .....	41-3
Summary of Database Settings .....	41-4
Precedence of Settings .....	41-4
Using Hyperion Essbase Application Manager for Database Settings .....	41-5
Using ESSCMD for Database Settings .....	41-7
Specifying Index Cache Size .....	41-8
Using the Hyperion Essbase Application Manager .....	41-8
Using ESSCMD .....	41-9
Making a Change Effective .....	41-9
Specifying Data File Cache Size .....	41-9
Using Hyperion Essbase Application Manager .....	41-10
Using ESSCMD .....	41-10
Making Changes Effective .....	41-10
Specifying Data Cache Size .....	41-11
Using Hyperion Essbase Application Manager .....	41-11
Using ESSCMD .....	41-12
Making Changes Effective .....	41-12
Enabling Cache Memory Locking .....	41-12
Using Hyperion Essbase Application Manager .....	41-13
Using ESSCMD .....	41-13
Making a Change Effective .....	41-13

Specifying Index Page Size .....	41-14
Using Hyperion Essbase Application Manager .....	41-14
Using ESSCMD .....	41-15
Making a Change Effective .....	41-15
Specifying Isolation Level .....	41-15
Notes on Isolation Level Options .....	41-16
Using Hyperion Essbase Application Manager .....	41-17
Using ESSCMD .....	41-18
Making Changes Effective .....	41-19
Specifying Disk Volumes .....	41-20
Using Hyperion Essbase Application Manager .....	41-22
Allocating Storage: An Example .....	41-24
Using ESSCMD .....	41-24
Making a Change Effective .....	41-26
Specifying Data Compression .....	41-27
Using Hyperion Essbase Application Manager .....	41-27
Using ESSCMD .....	41-28
Using SETDBSTATEITEM .....	41-28
Using SETDBSTATE .....	41-28
Making a Change Effective .....	41-29
Controlling Data Block Size .....	41-29

## Chapter 42: Ensuring Data Integrity

About Transactions .....	42-1
About Isolation Levels .....	42-2
About Committed Access .....	42-2
About Uncommitted Access .....	42-3
Handling a Transaction .....	42-4
Locking Data .....	42-5
Locking Under Uncommitted Access .....	42-6
Locking Under Committed Access .....	42-7
Concurrency Under Committed Access .....	42-9
Committing Data .....	42-9
Committing Data with Committed Access .....	42-10
Committing Data with Uncommitted Access .....	42-10
Accommodating Data Redundancy .....	42-11

Rolling Back Transactions .....	42-11
Rollback with Committed Access .....	42-11
Rollback with Uncommitted Access .....	42-12
Actions to Take When a Transaction Does Not Complete .....	42-12
Checking Structural Integrity .....	42-13

## **Part VIII: Designing and Building Currency Applications**

### **Chapter 43: Designing and Building Currency Conversion Applications**

Overview of the Business Problem .....	43-2
Overview of the Currency Application's Structure .....	43-2
Contents of the Main Database .....	43-3
Contents of the Currency Database .....	43-5
Conversion Methodologies .....	43-6
Steps for Creating a Currency Conversion Application .....	43-7
Creating the Main Database Outline .....	43-7
Open the Existing Database Outline .....	43-8
Modify the Measures Dimension .....	43-8
Modify the Market Dimension .....	43-11
Modify the Scenario Dimension .....	43-14
Save the Main Database Outline Changes .....	43-17
Creating the Currency Database Outline .....	43-17
Generate the Currency Database Outline .....	43-18
Review the Contents of the Currency Database Outline .....	43-21
Add New Members to the CurType Dimension .....	43-22
Save the Currency Database Outline Changes .....	43-23
Linking the Main and Currency Databases .....	43-24
Calculating Currency Conversions .....	43-26
Overwriting Local Values with Converted Values .....	43-26
Keeping Local and Converted Values .....	43-27
Converting Data to a Different Currency in Reports .....	43-29
Effects of CCTRACK Parameter on Conversion Calculations .....	43-30

## Part IX: Maintaining Hyperion Essbase Applications

### Chapter 44: Performing Interactive and Batch Operations Using ESSCMD

ESSCMD Basics .....	44-1
Syntax Guidelines .....	44-2
Quotation Marks .....	44-2
Semicolon Statement Terminator .....	44-2
Running ESSCMD on Different Operating System Platforms .....	44-3
Canceling ESSCMD Operations .....	44-3
Referencing Files .....	44-3
Multi-User Considerations .....	44-5
Case-Sensitivity .....	44-5
Getting Help .....	44-5
Quick Start to ESSCMD .....	44-5
Starting and Quitting ESSCMD .....	44-6
Using Interactive Mode .....	44-6
Logging Into the Server .....	44-7
Entering Commands .....	44-8
Canceling Operations .....	44-8
Using Script and Batch Files for Batch Processing .....	44-9
Writing Script Files .....	44-10
Running Script Files .....	44-10
Handling Command Errors in a Script File .....	44-11
Sample Script Files .....	44-12
Sample Script: Importing and Calculating Data .....	44-13
Sample Script: Building Dimensions and Importing/Calculating Data from a SQL Source .....	44-14
Sample Script: Scheduling Report Printing .....	44-15
Writing Batch Files .....	44-15
Handling Command Errors in Batch Files .....	44-16

## Chapter 45: Running Hyperion Essbase, Applications, and Databases

About the Agent .....	45-1
Starting and Stopping the Hyperion Essbase Server .....	45-2
Starting the Hyperion Essbase Server in the Foreground .....	45-2
Starting the Hyperion Essbase Server in the Background .....	45-2
Improving Parallel Login Processing .....	45-3
Stopping the Hyperion Essbase Server from the Agent .....	45-3
Stopping the Hyperion Essbase Server Remotely .....	45-3
Using ESSCMD to Shut Down the Server .....	45-3
Using an ESSCMD Batch Script to Shut Down the Server .....	45-4
Using the Server Console .....	45-4
Starting and Stopping Applications .....	45-5
Starting an Application .....	45-6
Stopping an Application .....	45-6
Starting and Stopping Databases .....	45-8
Starting a Database .....	45-8
Stopping a Database .....	45-8
Viewing a List of Users and Available Ports .....	45-10
Disconnecting a User From the Server .....	45-10
Changing the System Password .....	45-11
Displaying the Server Software Version Number .....	45-12
Viewing Security System Information .....	45-12
Shutting Down All Open Applications and Quitting Hyperion Essbase .....	45-12
Understanding Client-Server Communications .....	45-12
Hyperion Essbase Client Types .....	45-13
Flow of Communications Events .....	45-14
Multithreading .....	45-15
Application Monitoring .....	45-16

## Chapter 46: Using Diagnostics to Monitor Performance

Overview of Diagnostic Tools .....	46-2
Viewing Application Information .....	46-2
Viewing Server Information .....	46-3
Viewing License Information .....	46-4
Viewing Configuration Information .....	46-5
Viewing System Information .....	46-6
Viewing Disk Drive Information .....	46-7
Viewing Application Status Information .....	46-8
Viewing Database Information .....	46-9
Viewing General Database Information .....	46-10
Viewing Database Storage Information .....	46-11
Viewing Currency Database Information .....	46-12
Viewing Database Statistics .....	46-13
Viewing Run-Time Information .....	46-14
Viewing Database File Information .....	46-16
Viewing a Record of Database Modifications .....	46-17
Quick Reference to Diagnostic Tools .....	46-18
Exception Error Handling .....	46-26
Error Log Information .....	46-26
Error Log Names and Locations .....	46-27
Application Event Log File .....	46-28
Viewing the Application Log File .....	46-28
Deleting the Application Log File .....	46-29
Considering Performance .....	46-30
Server Event Log File .....	46-31
Viewing the Server Log File .....	46-31
Deleting the Server Log File .....	46-32
Creation of a Directory for Trace Files .....	46-33
Defining an ARBORDUMPPATH Environment Variable .....	46-34
Defining a DUMP Directory .....	46-34
Trace File Notification .....	46-35

Outline Change Log File .....	46-35
Outline Change Detail That is Displayed in the Log .....	46-36
Outline Change Log On and Off Function .....	46-38
Setting for the Outline Change Log File Size .....	46-39
Object Locks .....	46-40
Overriding a File Lock .....	46-40
Unlocking Objects .....	46-40
Application Monitoring .....	46-41
Server Error Message Categories .....	46-41

## **Chapter 47: Working with Hyperion Essbase Files and Cross-Platform Environments**

About Hyperion Essbase File Types .....	47-1
Operating on Applications, Databases, and Related Objects .....	47-5
Copying an Application .....	47-6
Renaming an Application .....	47-7
Deleting an Application .....	47-8
Copying a Database .....	47-9
Renaming a Database .....	47-10
Deleting a Database .....	47-11
Copying Objects .....	47-12
Renaming Objects .....	47-13
Deleting Objects .....	47-13
Locking and Unlocking Objects .....	47-14
Locking Objects.....	47-14
Unlocking Objects .....	47-15
Porting Applications Across Platforms .....	47-15
Identifying Compatible Files .....	47-15
Checking File Names .....	47-16
Transferring Compatible Files .....	47-18
Using Using FTP to Transfer .....	47-18
Using Hyperion Essbase Application Manager to Transfer .....	47-19
Redefining Server Information.....	47-19
Reloading the Database .....	47-20

## **Chapter 48: Backing Up Data and Recovering Databases**

Backing Up a Database .....	48-1
Archiving Data .....	48-2
Exporting Data .....	48-3
Using the EXPORT Utility .....	48-4
Reloading Exported Data .....	48-4
Recovering from a Crashed Database .....	48-5
What to Expect If a Server Interruption Occurs .....	48-6
Spreadsheet Update Logging .....	48-8

## **Appendix A: Performance-Related Settings**

### **Glossary**

### **Index**

Part VI describes how to create reports on the data in Hyperion Essbase OLAP Server databases, including how to create reports quickly, develop report scripts, optimize reports, and export data to other programs. Part VI contains the following chapters:

- Chapter 35, “Quick Start to Report Scripts,” introduces you to the basic concepts behind reports and describes how to create them using the Report Writer.
- Chapter 36, “Developing Report Scripts,” describes how to create complex report scripts, including page layout and formatting information, how to select and sort members, how to restrict and order data values, how to convert data to a different currency, and how to generate reports using the C, Visual Basic, and Grid APIs.
- Chapter 37, “Examples of Report Scripts,” contains detailed examples of report scripts.
- Chapter 38, “Optimizing Your Reports,” describes ways to generate your reports more quickly.
- Chapter 39, “Copying Data Subsets and Exporting Data to Other Programs,” describes how to move data from Hyperion Essbase OLAP Server databases to other programs by extracting an output file of the data to move using the Report Writer.



A Hyperion Essbase Application Manager report lets you retrieve formatted summaries from a database.

There are several ways that you can report on the data in your database:

- Use the Application Manager's Report Writer to create a report script and run your report, as explained in this chapter.
- Generate database reports through a spreadsheet. You can use report commands in a spreadsheet in Free-Form mode or in Template Retrieval mode. For information on reporting through your spreadsheet interface, see your *Hyperion Essbase Spreadsheet Add-in User's Guide*.
- Use Hyperion Essbase APIs. See Chapter 36, "Developing Report Scripts," for more information about using report APIs or the online *API Reference* in the DOCS directory for syntax and technical descriptions.
- Use third-party reporting tools such as Crystal Info for Hyperion Essbase (sold separately from Hyperion Essbase).

Use Report Writer when you need to create large-scale reports consisting of many pages of multidimensional data. Reports of this scale often can exceed the capabilities of even the most robust spreadsheet. You can use Application Manager to create report scripts and run reports, or you can schedule report scripts to run in batch mode, using the ESSCMD command-line interface. See Chapter 44, "Performing Interactive and Batch Operations Using ESSCMD" for information about ESSCMD.

Report Writer commands let you define formatted reports, export data subsets from an Hyperion Essbase database, and produce free-form reports.

This chapter provides fundamental information about reports and report scripts, including:

- “Creating a Simple Report Script” on page 35-2
- “Parts of Report Scripts and Reports” on page 35-5
- “Planning a Report” on page 35-10
- “Security and Multi-User Issues” on page 35-10
- “Creating and Editing Report Scripts” on page 35-11
- “Running Report Scripts” on page 35-21
- “Developing Free-Form Reports” on page 35-28

For information about creating complex report scripts, see Chapter 36, “Developing Report Scripts.”

## Creating 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 a report script specifies these elements, dimensions, and member selection commands. It includes comments, which document the behavior of the script, and the ! output command. This example is based on the Sample Basic database, which is supplied with your Hyperion Essbase OLAP Server installation.

- To create a simple report script:
1. Start Application Manager, and connect to your Hyperion Essbase server.
  2. Select the Sample application and the Basic database, and click the Report Scripts  button.

If you do not have the Sample Basic database installed, contact your Hyperion Essbase administrator.

3. Click the New Button to open the Report Editor.

4. Type the following information, with the exception of the commented (//) lines, which are for your reference:

```
// 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
<CHILDREN Market
Qtr1 Qtr2
Actual Budget Variance
<CHILDREN Product

// Finish with a bang
!
```

5. Choose File > Save, and type **Myrept1** for the report script object name, and save it on the server (the default).
6. Choose Report > Run.

When you run this report against the Sample Basic database, the script 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

For information about Report Writer command syntax and definitions, see the online *Technical Reference* in the DOCS directory. If you are using the Report Editor, choose the Help menu item to consult the Application Manager Help.



To create report scripts in ESSCMD:

Use the REPORTLINE command to execute a single report line.  
For example,

```
REPORTLINE "<DESCENDANTS Year !"
```

Use the REPORT command to execute one or more report lines.  
For example,

```
REPORT <IDESCENDANTS Year <ICHILDREN Market  
!
```

See the online *Technical Reference* in the DOCS directory for information about this command. See Chapter 44, “Performing Interactive and Batch Operations Using ESSCMD” for information about ESSCMD.

## Parts of Report Scripts and Reports

The Hyperion Essbase Report Writer consists of three main components:

- *Report Editor* is an ASCII text editor that you use to write the report script. The Report Editor features a text editing window and customized menus. Saved report scripts have the file extension `.REP`.
- *Report Extractor* retrieves the data information from the Hyperion Essbase database when you run a report script.

- *Report Viewer* displays the complete report. Saved reports have the file extension .RPT.

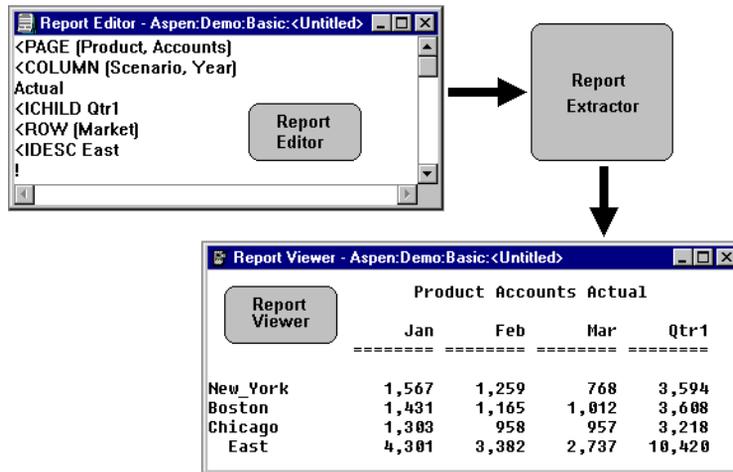


Figure 35-1: Report Writer Components

## How the Report Extractor Retrieves Data

The Report Extractor processes the report script and retrieves data in the following order:

1. Composes the member list, based on all possible member combinations. For example, the following command retrieves member East and all of its descendants:

```
<IDESCENDANTS East
```

2. Applies member restrictions. For example, the following command refines the member selection:

```
<LINK
```

3. Orders the member output. For example, the following command determines the order in which members are sorted:

```
<SORT
```

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:
 

```
{SUPPRESSMISSINGROWS}
```
6. Sorts data. For example, the following command returns rows with the highest values of a specified data column:
 

```
<TOP
```
7. Formats output. For example, the following command skips one or more lines in the final output report:
 

```
{SKIP}
```

The order in which the Report Extractor retrieves data is important when using complex extraction and formatting commands. For example, because the Extractor restricts data (step 5) before sorting data (step 6), if you place conditional retrieval commands in the wrong order, the report output results could be unexpected. Be aware of the data retrieval process when designing your report scripts.

## Parts of a Report

Understanding the parts of a report is essential as you plan and design your own reports. 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 ID 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, whereas 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.

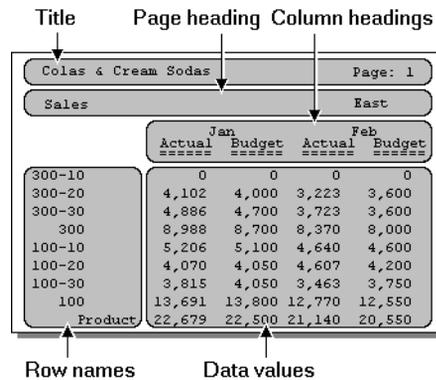


Figure 35-2: Elements of a Typical Report

## Parts of a Report Script

A report script consists of a series of Hyperion Essbase Report Writer commands, terminated by the bang (!) report output command.

You can enter one or more report scripts in a report script file. A report script file is an ASCII text file that you create with the Hyperion Essbase Report Editor or any text editor.

To build a report script, enter commands that define the layout, member selection, and format in the Hyperion Essbase Report Editor.

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 for 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 (<).

Additionally, the *bang* character (!) terminates a series of commands and requests information from the database. You must terminate a report script with a bang character, or you can use several bang characters within the script. See Chapter 38, “Optimizing Your Reports,” for more information about the ! character.

See the online *Technical Reference* in the DOCS directory for detailed information about the various report commands that you can use.

## Planning a Report

Report design is an important part of presenting your information. Designing a report is easy if you include the proper elements and arrange information in an attractive, easy-to-read layout.

- To plan your report:
  1. Consider your reporting needs and the time it will take to generate the report.
  2. Make a rough sketch of the report that includes:
    - Report layout
    - Number of columns
    - Members to include
    - Titles, if applicable
    - Format of the data values
  3. Review the sketch; if you need to add additional data or formatting to the report, this is often apparent at this stage.
  4. Determine ways to optimize the run time of the report. See Chapter 38, “Optimizing Your Reports,” for suggestions about optimizing your report script.

**Note:** As you plan your report, minimize your use of numeric row names. To avoid ambiguity, give the rows names that describe their content.

## Security and Multi-User Issues

Because you run the Report Editor from the Application Manager menu, you must have access to the Application Manager in order to use the Report Editor to create or modify a report script. You can also use any text editor to create script files. If you use the Application Manager’s Report Editor, it lets you create and modify report scripts stored on your desktop machine, as well as the Hyperion Essbase server. To modify report scripts stored on the server, you must have Application Designer or Database Designer access.

Hyperion Essbase supports concurrent, multi-user database access. As in most multi-user environments, Hyperion Essbase protects your critical data with a security system. Users can read or update data only if they have the correct privileges.

When you execute a report script, the Hyperion Essbase security system verifies that you have Read or higher access level to all data members specified in the report. In a filtering process identical to the one for retrieving members into a spreadsheet, Hyperion Essbase filters any member from the output for which you have insufficient privileges.

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 your report, the lock does not interfere with your 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 your last report ran.

See Chapter 17, “Managing Security at Global and User Levels” for more information about the Hyperion Essbase security system.

## Creating and Editing Report Scripts

You can create your report script in the Report Editor or with any ASCII text editor. Once you create your script, you can choose to save it to either the Hyperion Essbase server or your desktop machine.

You can modify your script using text editing features that let you cut, copy, paste, find, and replace. You can also undo most commands and changes with the Edit > Undo command.

The Report Editor uses familiar text editing commands, such as Edit > Cut, Edit > Copy, and Edit > Paste. These commands are available from the Application Manager menu or through accelerator key (hot key) combinations.

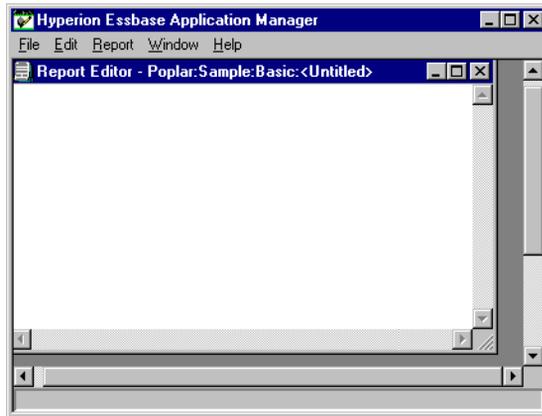
## Creating New Report Scripts

Before you can create a report script, you must connect to an Hyperion Essbase server and open a database.

► To create a new report script:

1. Choose File > New > Report Script from the Application Manager menu.

The Report Editor displays.



*Figure 35-3: New Report Editor Window*

2. Begin typing your report script.

The name of the report script is Untitled when it is first displayed. When you save the script, the Application Manager prompts you for a file name.

## Saving Report Scripts

You can save a report script as:

- A file on your client machine.
- An object on the Hyperion Essbase server. If you want other users to have access to the report script, you need to save it on the Hyperion Essbase server. You can associate the script object with:

- An application and all the databases within the application, which lets you run the script from any of the databases in the application.
- A database, which lets you run the script from the specified database.

Report scripts have a `.REP` extension by default. If you run a report script from the Application Manager it must have a `.REP` extension.

➤ To save a report script:

1. Choose File > Save.
2. Choose to save the script on the Hyperion Essbase server or to your desktop machine.

➤ To save the script on the Hyperion Essbase server:

1. Click the Server button in the Location box.
2. Type the name to give to the file in the Object Name box.
3. Choose the server where you want to save the script in the Server list box.
4. Choose the application where you want to save the script in the Application list box.
5. Choose the database where you want to save the script in the Database list box, or choose “(all dbs)” to make the report script available to all databases within the chosen application.
6. Choose Report Scripts in the List Files of Type box.
7. Click OK.

- To save the script on your desktop machine:
  1. Type the file name in the Object Name box.
  2. Choose the application where you want to save the script in the Application list box.
  3. Choose the database where you want to save the script in the Database list box, or choose “(all dbs)” to make the report script available to all databases within the chosen application.
  4. Choose Report Scripts in the List Files of Type box.
  5. Click OK.

By default, the file is saved to the `\ESSBASE\CLIENT\SAMPLE` directory on your desktop machine.

## Opening Report Scripts from the Application Manager

- To open a report script from the Application Manager:
  1. From the Application Manager menu, choose File > Open.

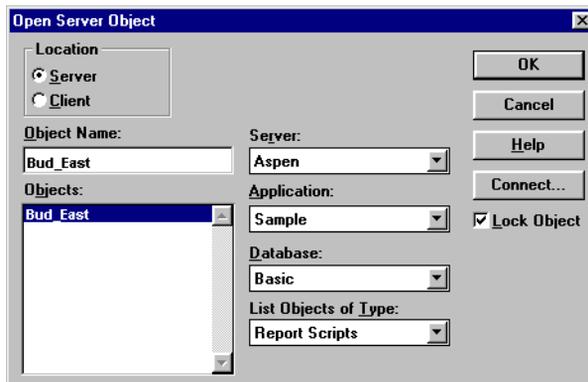


Figure 35-4: Open Server Object Dialog Box

2. From the Location group, select the storage location for the report script.
  - Click the Server option to search the Hyperion Essbase server.
  - Click the Client option to search your desktop machine. The dialog box title changes to Open Client Object.

3. From the List Objects of Type list box, choose Report Scripts.

The Objects list box displays the names of all available report scripts in the selected application and database.

**Note:** To choose another application to search, select it from the Application list box. To choose another database to search, select it from the Database list box.

4. You can also locate a locally stored report script through the Windows file system. Click the File System button in the Client Object dialog box.

The Open Client File dialog box is displayed. The file name mask \* .REP is displayed in the File Name text box and all report script files in the current directory are displayed in the list box.

5. To search other directories, use the Directories list box.
6. When the report script you want to open is displayed in the Objects list of the Open Server Object, Open Client Object, or Open Client File dialog box, you can:
  - Click the report script name. When the name is displayed in the Object Name text box, click OK.
  - Enter the name in the Object Name text box and click OK.
  - Double-click on the report name in the Objects list.

The Application Manager starts the Report Editor and loads the selected report script.

## Opening Report Scripts from an Application or Database Directory

You can open report scripts stored in applications and databases from the application directory.

- To open a report script from an application folder:
  1. Double-click the application name to open the application folder and select a database from the Databases list that contains a report script, or select “(all dbs)” from the Databases list if you want to open a report script stored at the application level.
  2. Click the Report Scripts button.
  3. From the Reports list, choose the report script you want to open.
  4. Click Open to start the Report Editor and load the selected report script.



Figure 35-5: Database with a Report Script

## Finding Text

You can use the Find menu command to search for text in the Report Editor. Find locates all occurrences of the specified text in your report script.

- To find text in a report script:
  1. Choose Edit > Find to open the Find dialog box.



Figure 35-6: Find Dialog Box

2. Type the text you want to find in the “Find what” text box.
3. Use the Direction group box to choose the search direction.
  - Click the Down option button to search from the cursor position forward to the end of the report script.
  - Click the Up option button to search backward from the cursor position to the beginning of the report script.

For example, Figure 35-6 shows a dialog box that finds the next occurrence of <CHILDREN “400” between the current cursor position and the end of the report script.

4. If you want the search to be case-sensitive, check the “Match case” check box. The search matches uppercase and lowercase letters exactly as they are displayed in the “Find what” text box.

In the example, the “Match case” check box is checked, so the search finds <CHILDREN “400” but not <Children “400”.

5. Click “Find Next.”

If the search finds the text you entered in the “Find what” text box, it highlights the text in the Report Editor.

6. To edit the selection, click Cancel to close the Find dialog box.

7. To search for other occurrences of the text, click Find Next again.

When Application Manager reaches the end of the report script, or if it cannot find the selected text to replace, the following dialog box is displayed:

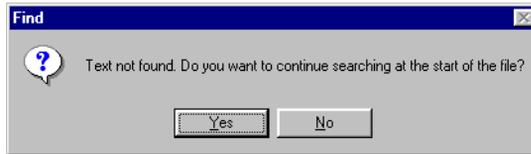


Figure 35-7: Continue Find Dialog Box

- Click Yes to continue searching the file.
  - Click No to close the dialog box and return to the Find dialog box.
8. When you are finished, click Cancel to close the Find dialog box.

## Replacing Text

The Replace menu command replaces one or all instances of specified text with different text.

- To replace text in a report script:
  1. Choose Edit > Replace to open the Replace dialog box.

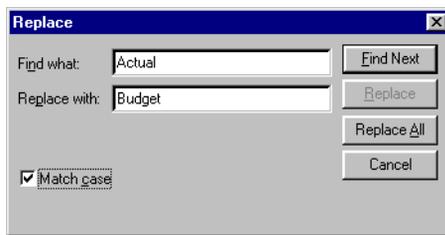


Figure 35-8: Replace Dialog Box

2. Enter the text you want to find in the “Find what” text box.

If you want the search to be case-sensitive, check the “Match case” check box. The search matches uppercase and lowercase letters exactly as they are displayed in the “Find what” text box.

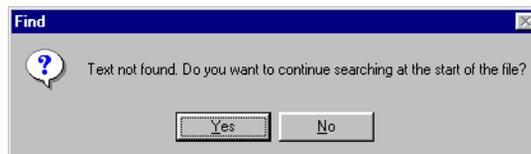
In the example, the “Match case” check box is checked, so the search finds Actual but not ACTUAL.

3. Enter the text in the “Replace with” box that will replace one or all instances of the contents of the “Find what” box.
4. Click “Find Next.”

If the search finds the text you entered in the “Find what” text box, it highlights the text in the Report Editor.

5. Click Replace to replace the highlighted text with the information in the “Replace with” box.
6. Click Replace All to replace all instances of the text in the “Find what” box with the information in the “Replace with” box.

When the search reaches the end of the report script, or if it cannot find the selected text to replace, the following dialog box is displayed:



*Figure 35-9: Continue Find Dialog Box*

- Click Yes to continue the searching the file.
  - Click No to close the dialog box and return to the Replace dialog box.
7. When you are finished, click Cancel to close the Replace dialog box.

## Cutting, Copying, and Pasting Text

Cutting, copying, and pasting are basic text editing features that you use to modify your report script. These features work with the desktop Clipboard, allowing you to take information from the Report Editor (by copying or cutting) and then bring it back into the Report Editor (by pasting). You can copy, cut, and paste between different scripts in the Report Editor window. For example, you can copy text from Script1, open Script2, and paste the text into Script2.

➤ To copy text:

1. In the Report Editor window, select the text to copy.
2. Choose Edit > Copy.

The text is still displayed in the Report Editor window, and a copy of the information is stored in the desktop Clipboard.

➤ To cut text:

1. In the Report Editor window, select the text to cut.
2. Choose Edit > Cut.

The text is removed from the Report Editor window, and is stored in the desktop Clipboard.

➤ To paste text:

1. In the Report Editor window, place your cursor where you want the text to be displayed.
2. Choose Edit > Paste.

The Paste command is disabled if there is nothing stored in the Clipboard.

## Deleting Text

➤ To remove text from the Report Editor:

1. In the Report Editor window, select the text to delete.
2. Choose Edit > Delete, or use the Delete key on your keyboard.

# Running Report Scripts

The Report menu in Report Editor features commands that you can use to choose a destination for the report, select a database against which to run the report script, and run the report script to generate a final report. The following sections describe these commands.

## Choosing the Report Output

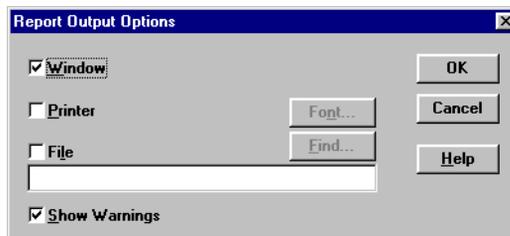
The Output Options menu command opens the Report Output Options dialog box that lets you choose three output destinations for your report:

- **Window**—displays your report in a window using the default system font.
- **Printer**—sends your report to the currently selected printer and lets you choose the font to use for the report. If Window is also selected, window output is displayed in the same font.
- **File**—sends your report to a .RPT file.

You can select as many output options as you like, but you must select at least one.

## Sending the Report Output to a Window

- To send your report output to a window:
  1. Choose Report > Output Options from the Report Editor.



*Figure 35-10: Report Output Options Dialog Box with Window Check Box Selected*

The Window check box is selected as the default.

If you also check the Printer check box, you can select a printer font for Hyperion Essbase to use as the screen font. See “Sending Report Output to a Printer” on page 35-22 for details.

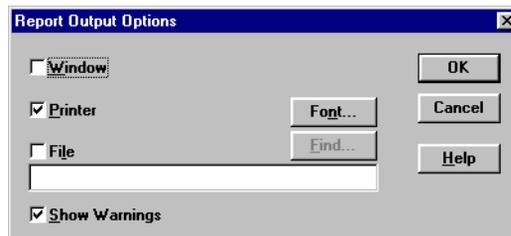
2. The Show Warnings check box is selected as the default. This option displays any warning messages about report processing in a window on the screen. If you want to turn off this display, click the check box.
3. To display your report in a window, click OK.

The Report Output Options dialog box closes. You are ready to select a database and run the report.

**Note:** You must check at least one output option to run a report, and you can check any combination of output options.

## Sending Report Output to a Printer

- To send your report output to a printer:
1. Choose Report > Output Options from the Report Editor menu.
  2. Click the Window check box to deselect it if you do not want the report to be displayed in a window.
  3. Check the Printer check box. Application Manager enables the Font button.



*Figure 35-11: Report Output Options Dialog Box with Printer Check Box Selected*

4. Click the Font button to select a printer font for the report.

The fonts, font styles, and font sizes available on your system are shown in the Font dialog box.

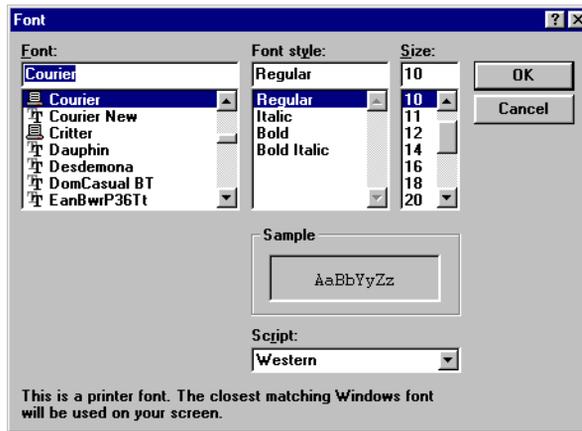


Figure 35-12: Font Dialog Box

5. From the Font list, select the font type to use in the report.
6. If you want to change the style of the font, choose a different style from the Font Style list.
7. From the Size list, choose the point size of the font. A sample of the font is displayed in the Sample box.
8. When you have finished configuring font options, click OK.

The Font dialog box closes, and you return to the Report Output Options dialog box.

9. In the Report Output Options dialog box, the Show Warnings check box is selected as the default. This option displays any warning messages about report processing in a window on the screen. If you want to turn off this display, click the check box.

10. Click OK to prepare the report for the printer.

The Report Output Options dialog box closes. You are ready to select a database and run the report.

**Note:** You must check at least one output option to run a report, and you can check any combination of output options.

## Sending Report Output to a File

- To send your report output to a file:
  1. Choose Report > Output Options from the Report Editor.
  2. Click the Window check box to deselect it if you do not want the report to be displayed in a window.
  3. Check the File check box.

Application Manager enables the Find button.



*Figure 35-13: Report Output Options Dialog Box with File Check Box Selected*

4. If you want to save the report to your desktop file system, click Find to choose a directory location.

The file name mask for report output, \* .RPT, is displayed in the “File name” text box. Any report scripts already in the current directory are displayed in the list box.

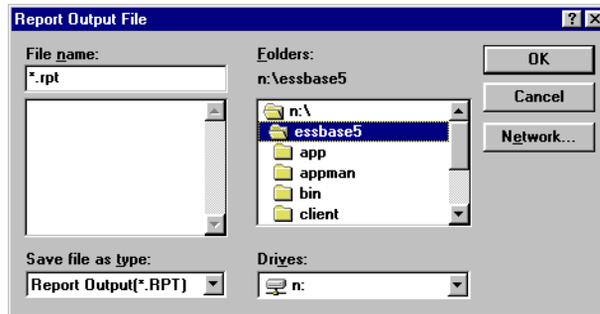


Figure 35-14: Report Output File Dialog Box

5. In the “File name” text box, enter a name for the report script, while retaining the .RPT extension.

Report script files follow the same naming conventions as DOS files.

6. In the Folders list box, select the directory location for the report file.
7. Click OK to save the name and location. The Report Output File dialog box closes and returns to the Report Options dialog box. The File text box displays the file name and directory location of the report script.
8. Click OK to prepare the report script.
9. The Show Warnings check box is selected as the default. This option displays any warning messages about report processing in a window on the screen. If you want to turn off this display, click the check box.

The Report Output Options dialog box closes. You are ready to select a database and run the report.

**Note:** You must check at least one output option to run a report, and you can check any combination of output options.

## Choosing Report Databases

The Select Database dialog box lets you select a database against which to run your report script.

- To choose a report database:
  1. Choose Report > Database from the Report Editor.

The Select Database dialog box is displayed.

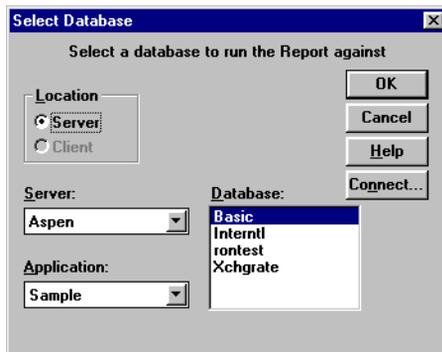


Figure 35-15: Select Database Dialog Box

2. Click the Server or Client option to specify the location of the database.
3. From the Server list box, select the server where the database resides.  
If there are no server names in the list, click the Connect button to log onto the server.
4. Select the application where the database resides from the Application list box.  
Application Manager lists the databases that reside on the selected server and in the selected application.
5. Select the database from the Database list.
6. Click OK.

When you run the report, it reports on the selected database.

**Note:** If the report is already associated with a database or application other than the one you specify, the `Select > Database` command does not change that association but simply overrides the default database for this Report Editor session only.

## Running the Report

The `Report > Run` menu command sends your report to the screen and any other output destination you have selected. See “Choosing the Report Output” on page 35-21 to learn how to choose a report output.

To run the report script, choose `Report > Run`.

## Running Multiple Report Script Files



You can use the `RUNREPT` command in `ESSCMD` to run one or more report script files. See the online *Technical Reference* in the `DOCS` directory for information about this command. See Chapter 44, “Performing Interactive and Batch Operations Using `ESSCMD`” for information about `ESSCMD`.

35

## Executing a Report String Without Creating a Report Script



You can use the `REPORTLINE` command in `ESSCMD` to run a single report string. See the online *Technical Reference* in the `DOCS` directory for information about this command. See Chapter 44, “Performing Interactive and Batch Operations Using `ESSCMD`” for information about `ESSCMD`.



You can use the `REPORT` command in `ESSCMD` to run multiple report strings. See the online *Technical Reference* in the `DOCS` directory for information about this command. See Chapter 44, “Performing Interactive and Batch Operations Using `ESSCMD`” for information about `ESSCMD`.

## 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 Application Manager's Report 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. The rest of the report is automatically produced in a format similar to that of any other report. When PAGE, COLUMN, and ROW commands are omitted, Hyperion 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 any 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
!
```

The Report Extractor applies free-form formatting rules to this report as follows:

1. Because California, Oregon, Washington, Utah, Nevada, and West are listed last, the Report Extractor treats them as if ROW (Market) had been specified (according to 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 (according to 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), according to rule 3.
4. The Report Extractor searches the Database Outline and finds that dimension Product is not specified in the report specification. Since Product is a single-member selection, the Report Extractor treats this member as if PAGE (Product) had been specified (according to 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.

Once 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 some of the data values' display. Formatted data values are displayed in the report when you run the script, based on the combined extraction and report 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:

- Let you customize the format and appearance of a report and create report-time calculations. Formatting commands are generally enclosed inside left and right braces ( { } ), although there are several formatting commands that begin with the < character.
- Are either applied globally within the report script or are specific to a member.

This chapter provides information about creating complex report scripts, including:

- “Syntax Guidelines” on page 36-2
- “Designing the Page Layout” on page 36-3
- “Formatting” on page 36-8
- “Selecting and Sorting Members” on page 36-27
- “Restricting and Ordering Data Values” on page 36-45
- “Converting Data to a Different Currency” on page 36-51
- “Generating Reports Using the C, Visual Basic, and Grid APIs” on page 36-52

For fundamental information about reports and report scripts, see Chapter 35, “Quick Start to Report Scripts.”

## Syntax Guidelines

To build a report, you enter commands that define the layout, member selection, and format you want in the Application Manager’s Report Editor. When you write a report script, follow these guidelines:

- Separate commands with at least one space, tab, or new line. Report processing is not affected by extra blank lines, spaces, or tabs.
- Enter commands in either 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 a single 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”).
  - Names containing one or more numerals at the beginning of the name (for example, “100-Blue”).
  - Names containing any of the following characters:
    - \* (asterisks)
    - @ (at signs)
    - [ ] (brackets)
    - :
    - ,
    - { } (braces)
    - (dashes, hyphens, or minus signs)
    - < (less than signs)
    - ( ) (parentheses)
    - + (plus signs)
    - / (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}.
  
- Use // (double slash) 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 multi-line comments.

## 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.

For information about formatting the page, column, or row headings, see “Formatting Page, Column, and Row Headings” on page 36-10.

## Creating Page, Column, and Row Headings

➤ To design the report headings:

1. Type <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. For example,

```
<PAGE (Measures, Market)
```

2. Press Enter.

3. Type <COLUMN (*dimensionname*), where *dimensionname* equals the name of each dimension to display across the page. For example,

```
<COLUMN (Year)
```

You can add dimension names to create nested column headings.

4. Press Enter.

5. Type <ROW (*dimensionname*), where *dimensionname* equals the name of each dimension to display down the page. For example,

```
<ROW (Market)
```

6. Press Enter.

**Note:** You can select additional members to associate with the heading commands. If you type a member name as a parameter for the PAGE, COLUMN, or ROW commands, Report Extractor automatically associates the member with the appropriate dimension.

The following report script is based on the Sample Basic database:

```
<PAGE (Product, Measures)
<COLUMN (Scenario, Year)
Actual
<CHILDREN Qtrl
<ROW (Market)
<IDESCENDANTS East
!
```

This script produces the following report:

Product Measures Actual				
	Jan	Feb	Mar	Qtrl
	=====	=====	=====	=====
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
<CHILDREN 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:

To Do This...	Use the 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 non-suppressed 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 information about suppressing the display of headings in the report, see “Suppressing Page, Column, and Row Formatting” on page 36-14.

## Creating Symmetric and Asymmetric Reports

Hyperion Essbase reports can contain symmetric or asymmetric column groups. Hyperion 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									

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, Hyperion Essbase creates a symmetric report unless you select the same number of members for all column dimensions.

See Chapter 37, “Examples of Report Scripts” for an example of an asymmetric report.

The Hyperion Essbase evaluation of symmetry versus asymmetry takes place prior to 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 Hyperion 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 only need to change 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 like 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 like those used in symmetric reports. An asymmetric report uses the <BLOCKHEADERS mode of column layout.

## Formatting

Formatting commands define the format of data and labels in the final report. These commands are generally enclosed in left and right braces ( { } ).

The two types of formatting commands are *global* and *member-specific* commands.

- 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 a additional rows to skip lines, you must use the SKIP command again.

## Formatting Report Pages

There are a number of formatting commands that you can use to design the look of your final report pages.

See Chapter 37, “Examples of Report Scripts” for report formatting examples.

### Setting the Page Length, Width, and Centering

You can set the following page specifications in the report script:

To Do This...	Use the Report Command
Specify the column widths in a report.	WIDTH
Set the left margin of the report.	LMARGIN
Set the center of the page.	SETCENTER

## Inserting Page Breaks

You can set the following types of page breaks in the report script:

To Do This...	Use the 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 only two columns are set up, Actual and Budget. The {DECIMAL} command, however, refers to a column three, which does not yet exist. Hyperion 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).

Hyperion 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)

## Accommodating Long Column and Row Names

Member names that are too long to fit into the column are automatically truncated; the tilde character (~) signifies that part of the name is missing. Long member names are common when using aliases in the report.

There are a several ways to modify your columns to display the entire member name.

To Do This...	Use the Report Command
Define the column width for all row members in the column.	NAMEWIDTH
Change where the row member column is displayed, and to shift the remaining columns left or right to make room for the long member names.	NAMECOL

## Suppressing Page, Column, and Row Formatting

You can suppress the display of page heading, columns, and rows in your report by using various SUPPRESS commands.

To Suppress the Display of...	Use the Report Command
The default column heading in the report.	SUPCOLHEADING
Rows that have only zero or missing values.	SUPEMPTYROWS
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, format settings, and so forth. Use the OUTPUT command to reverse the actions of SUOUTPUT.	SUOUTPUT
Currency information when you use the CURRENCY command to convert the data values in your report to a specified currency.	SUPCURHEADING

## Repeating Row Names

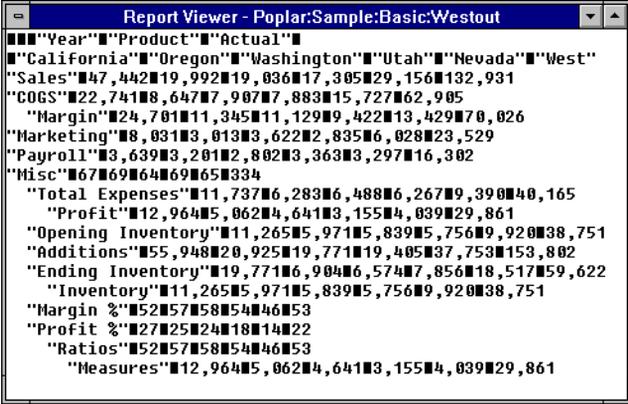
To repeat the row member names on every line of the report, use the <ROWREPEAT command. Use the <NOROWREPEAT command to prevent row member names from being repeated on each line of the report if the row member name does not change on the next line. NOROWREPEAT is enabled by default.

## Formatting Reports with Tab Delimiters

You can place tabs between columns rather than spaces in your report scripts. This is useful when you want to export report output into another form.

To replace spaces with tab delimiters, type {TABDELIMIT} anywhere in the report script.

When you save the report script, Hyperion Essbase automatically replaces the spaces with tabs. When you view the report in the Report Viewer, black squares indicate tab marks.



```

Report Viewer - Poplar:Sample:Basic:Westout
Year Product Actual
California Oregon Washington Utah Nevada West
Sales 47,442 19,992 19,036 17,305 29,156 132,931
COGS 22,741 8,647 7,907 7,883 15,727 62,905
Margin 24,701 11,345 11,129 9,422 13,429 70,026
Marketing 8,031 3,013 3,622 2,835 6,028 23,529
Payroll 3,639 3,201 2,802 3,363 3,297 16,302
Misc 67 69 64 69 65 334
Total Expenses 11,737 6,283 6,488 6,267 9,390 40,165
Profit 12,964 5,062 4,641 3,155 4,039 29,861
Opening Inventory 11,265 5,971 5,839 5,756 9,200 38,751
Additions 55,948 20,925 19,771 19,405 37,753 153,802
Ending Inventory 19,771 6,904 6,574 7,856 18,517 59,622
Inventory 11,265 5,971 5,839 5,756 9,200 38,751
Margin % 52 57 58 54 46 53
Profit % 27 25 24 18 14 22
Ratios 52 57 58 54 46 53
Measures 12,964 5,062 4,641 3,155 4,039 29,861

```

Figure 36-1: Tab-Delimited Report

## 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 Chapter 37, “Examples of Report Scripts.”

### Totaling Columns

The CALCULATE COLUMN command lets you create a new report column, perform on-the-fly calculations, and display the calculation results in the newly created column.

The following table summarizes column calculation commands:

To Do This...	Use the Report Command
Create a new 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 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 the standard mathematical operations. For syntax and parameter descriptions, see the online *Technical Reference* in the DOCS directory.

If you use the same name for more than one column, Hyperion 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 could, for instance, 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 two or more 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 do this, insert the tilde character (~) 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	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 not defined.

In cases where there are fewer column header dimensions than the number of levels that you want, you can create multi-line 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 online *Technical Reference* in the DOCS directory.

## Numbering Columns

If the number of regular (non-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. For example:

- If the first section of a report has 12 columns (including row name columns), and 3 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 6, then the regular columns are columns 0–5, and the same calculated columns are columns 6–8.
- Likewise, 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 columns names. The column numbering for a report with two different sections with varying numbers of regular columns would be:

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 would occur 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:

To Do This...	Use the Report Command
Create a new row and associate it with a row name or label. This 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 new 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 online *Technical Reference* in the DOCS directory.

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.
- Hyperion Essbase does not support nested (parenthetical) expressions.
- Hyperion 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.

If you specify an operator, it applies to subsequent output rows and stores the result in the calculated row. This 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 online *Technical Reference* in the DOCS directory.

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

You can use a variety of formatting commands to customize how data displays in your final report.

### Underlining

Use underlining as a visual aid to break up blocks of information in a report.

To Do This...	Use the Report Command
Set the default underline character that will display in the report.	UNDERLINECHAR
Underline all non-blank characters 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 your final report by using various SUPPRESS commands.

To Suppress the Display of...	Use the 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 36-14 for information about suppressing the display of page, column, and row formats.

## Indenting

Use indenting to provide visual clues to row levels of the script.

To Do This...	Use the 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 such valuable information about a report as:

- A date and time stamp
- The user ID 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 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 online *Technical Reference* in the DOCS directory 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 any text string that you want to display in the data cells.

You can place the MISSINGTEXT command at any point 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 36-23 for information about suppressing labels, or see the online *Technical Reference* in the DOCS directory 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 any text string that you want to display in the data cells.

## Adding Blank Spaces

Adding blank spaces in a report draws the reader to key information, such as totals.

To Do This...	Use the 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 your final report:

To Do This...	Use the 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

To Do This...	Use the Report Command
Use the European method for displaying numbers where decimal points are used as the thousands separator character while commas separate the decimal portion of the number from the integer portion (2.000,01, whereas 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 the selection and sorting of 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 your report, unless you change the member name on which the member selection command is based. Attribute dimensions can be included in member selection commands.

To Do This...	Use the Report Command
Select all the members from the same dimension as the dimension member.	ALLINSAMEDIM
Include all the siblings of the specified member.	ALLSIBLINGS
Include all the ancestors of the specified member.	ANCESTORS
Select a base dimension member based on its attributes.	ATTRIBUTE
Select all members in the level immediately below the specified member.	CHILDREN

<b>To Do This...</b>	<b>Use the Report Command</b>
Include the descendants of the specified member to the report, excluding the dimension top.	DESCENDANTS
Select the Level 0 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 all 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 all members from the same dimension and generation as the specified member.	OFSAMEGEN
Include all 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 Using Generation and Level Names

*Generation and level name* selection commands identify a specific level or generation of members, based on either:

- 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 your report. You can either define your own generation and level names, or you can use the default names provided by Hyperion Essbase. For information on generations and levels, see Chapter 13, “Introducing Dynamic Dimension Building.”

Using generation or level names whenever possible makes your report easier to maintain. Because you do not have to specify a member name in your report, you do not need to change the report if the member name is changed or deleted from the database outline.

**Note:** Generation and level names are stand-alone commands. You cannot use them in place of member names in report extraction or formatting commands. For example, you *cannot* use them with the <DESCENDANTS or <CHILDREN commands.

### 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 following format:

*Levn, DimName*

where *n* is the level number.

*DimName* is the name of the dimension from which you want to select the members.

**Note:** Do *not* leave 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 following format:

*Gen $n$ , DimName*

where  $n$  is the generation number.

*DimName* is the name of the dimension from which you want to select the members.

**Note:** Do *not* leave 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 the Outline Editor.

The following example is based on the Sample Basic database. It 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
!
```

The report script produces the following 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 Dynamic Time Series Members

You create and identify dynamic members in the database outline; they are members that are calculated only during user retrieval requests, such as generating a report script. The Time dimension contains a special Dynamic Time Series tag which has reserved 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
Week	W-T-D	Week-to-Date
Day	D-T-D	Day-to-Date

See Chapter 7, “Creating Applications and Databases” for information about creating and maintaining Dynamic Time Series generation names in the database outline.

**Note:** The outline’s database header message 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 of the following formats:

```
<LATEST memberName
```

where *memberName* is the name of the member in the Time dimension.

The <LATEST command is a global command that 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 type the Dynamic Time Series string exactly as it is displayed in the database outline; you cannot create your own 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 Using Boolean Operators

Boolean operators let you specify precise member combinations within your report, which is particularly useful when dealing with large outlines. Use the AND, OR, and NOT Boolean operators, combined with extraction commands, to refine your member selections within the report script.

- Use the AND operator when all conditions must be met.
- Use the OR operator when one condition of several must be met.
- Use the NOT operator 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 following 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:** You must select members from the same dimension, and all extraction command arguments must be enclosed in parentheses, as in the example above. NOT can only be associated 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 online *Technical Reference* in the DOCS directory for a list of all valid extraction commands that can be used in conjunction with the LINK command.

Examples:

```
<LINK (( <IDESCENDANTS("100") AND <UDA(Product,Sweet)) OR <LEV
(product,0))
```

selects sweet products from the “100” sub-tree, plus all level 0 products.

```
<LINK (( <IDESCENDANTS("100") AND NOT <UDA (Product,Sweet)) OR
<LEV (product,0))
```

selects non-sweet products from the “100” sub-tree, plus all level 0 products.

See Chapter 37, “Examples of Report Scripts” for other examples of narrowing member selection criteria.

## Selecting Members Using Substitution Variables

Substitution variables act as global placeholders for information that changes regularly; you set the substitution variables on the server through Application Manager or ESSCMD, and assign a value to each variable. You can then change the value at any time, reducing manual changes to a report script. You do need Database Designer privileges to set substitution variables.

For example, many reports are dependent on reporting periods; if you generate a report based on the current month, you would have to 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. Hyperion Essbase dynamically updates the information when you run the final report.

See Chapter 7, “Creating Applications and Databases” for information about creating and changing substitution variables in the database outline. See the online *Technical Reference* in the DOCS directory, 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:**

The substitution variable must be accessible from the application and database against which you are running the report.

At the point in the script where you want to use the variable, use the following format:

*&variablename*

where *variablename* is the same as the substitution variable set on the server.

For example,

```
<ICHILDREN &CurQtr
```

becomes

```
<ICHILDREN Qtr1
```

**Note:** The variable name can be an alphanumeric combination up to 80 characters in length, including underscore characters. You cannot use spaces or punctuation in the variable name.

When you run the report script, Hyperion Essbase replaces the variable name with the substitution value and that information is incorporated into the final report.

## Selecting Members 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 two or more 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 flavored juice and orange flavored 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 following format:

```
<ATTRIBUTE (mbrName)
```

where *mbrName* is the name of an attribute dimension member; for example:

```
<ATTRIBUTE (Bottle)
```

returns all products packaged in bottles.

There can be cases where attribute dimensions have members with the same name. For instance, the attribute dimension Ounces and the attribute dimension Age can each have a member named 24. To ensure that a query returns correct results, you must specify the full attribute dimension member name, as set in the Attribute Member Name dialog box in the Hyperion Essbase Application Manager. The following format:

```
<ATTRIBUTE (Ounces_24)
```

returns all products that are packaged in 24 oz. units.

Attribute types can be text, numeric, date, and Boolean. See Chapter 10, “Working with Attributes” for more information.

## Selecting Members Associated with a Specific Attribute

You can select all the 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 (AttrDimName, "Operator", Value)
```

For example:

```
<WITHATTR (Population, "IN", Small)
```

returns all base dimension members that are associated with the attribute Small from the Population attribute dimension.

```
<WITHATTR (Ounces, "<", 32)
```

returns all base dimension members that are associated with the attribute 32 from the Ounces attribute dimension.

**Note:** The <WITHATTR command can be used within the LINK command to refine member selections. For example:

```
<LINK ((<WITHATTR (Ounces, "<", 32) AND <WITHATTR ("Pkg Type",
"=", Can))
```

For more information on the syntax for the <ATTRIBUTE and <WITHATTR commands, see the online *Technical Reference* in the DOCS directory.

## Selecting Members by a Specific 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:

```
<WITHATTR ("Intro Date", "=", <TODATE ("mm-dd-yyyy",
"12-10-1996")
```

returns data on all products that were introduced *on* December 10, 1996.

The following format:

```
<WITHATTR ("Intro Date", "<", <TODATE ("mm-dd-yyyy",
"12-10-1996")
```

returns data on all products that were introduced *before* December 10, 1996.

The following format:

```
<WITHATTR ("Intro Date", ">", <TODATE ("mm-dd-yyyy",
"12-10-1996")
```

returns data on all products that were introduced *after* December 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).

For more information on the syntax for the <TODATE command, refer to the Report Writer commands in the *Quick Technical Reference*.

## Selecting Members Using User-Defined Attributes

A user-defined attribute (UDA) enables you to select and report on data based on a common characteristic. These attributes are particularly useful when performing member selections from an outline with an unbalanced hierarchy (a hierarchy where 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 Designer privileges 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. Hence, for data analysis, UDAs are not as powerful as attributes.

You can use the UDA command in conjunction with Boolean operators to refine report queries further. See “Selecting Members Using Boolean Operators” on page 36-32 for more information.

See Chapter 7, “Creating Applications and Databases” 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 following format:

```
<UDA (dimName, "UDAstring")
```

where *dimName* is the dimension of the member that you select, and *UDAstring* is the UDA that is set on the server. For example:

```
<UDA (product, "Sweet ")
```

When you run the report script, Hyperion 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 your own UDA string and incorporate it into the report script.

## Selecting Members Using Wildcards

You can use wildcards to select either members, generation, or level names in a report script. If you use member names, Hyperion Essbase searches the member and all descendants of that member. If you specify a generation or level name, Hyperion Essbase searches only members of that generation or level.

Using wildcards reduces the amount of member information needed for a script and simplifies script maintenance.

The following two types of wildcards are supported in Report Writer:

- Trailing asterisk (\*) characters at the end of the string to search for common member properties
- Pattern-matching question mark (?) characters 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 following format:

```
<MATCH mbrName, "charName*"
```

where *mbrName* is the name of the member that you select, and *charName* 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 following format:

```
<MATCH (mbrName, "???charNames")
```

where *mbrName* is the name of the member to select, and *charNames* 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 are used to represent the variable three characters in the string. If only two question marks were used in the example, no matches would be found. You can place question mark wildcards anywhere in the match string.

## Selecting Members Using Static Member Names

*Static member names* are non-changing member names, such as Sales and COGS, that you enter directly into the report script. Although they are easy to establish, static member name definitions can be difficult to maintain if your database outline changes. You must change the member name in every script that is used with that database.

In general, report scripts are easier to maintain if you use generation and level names, or member selection commands, rather than static member names whenever possible. For example, when you remove a product, such as Widgets,

from the dimension, the member selection command <children Product works, but a specific list referencing Widgets as a static member name generates the following error message:

```
"Unknown Member [Widgets]."
```

Static members tend to be particularly difficult to maintain when dealing with distributed OLAP databases, where both source and target databases must be consistent.

**Note:** These user-defined, static member generation and level names are not displayed in the Outline Editor.

The following report script uses a static member called ProductGroups to select all the Level 1, Product member names from the Sample Basic database.

```
<PAGE(Year)
<COLUMN(Product)
<ROW (Measures)
Qtr1
ProductGroups
Sales Profit
!
```

The report script produces the following report:

	Qtr1 Market Scenario				
	100	200	300	400	Diet
	=====	=====	=====	=====	=====
Sales	25,048	26,627	23,997	20,148	25,731
Profit	7,048	6,721	5,929	5,005	7,017

When you run a report that includes static member definitions, the report displays members in order of their definition in the report script by member name. Sort commands have no effect on static member definitions. See “Sorting Members” on page 36-44 for more information about the effects of sorting members.

## Suppressing Shared Members

You can suppress the display of duplicate shared members when you extract data for your report. You can only suppress shared member display in conjunction with the following:

- Generation names
- Level names
- DIMBOTTOM command
- OFSAMEGEN command
- ONSAMELEVELAS command

Suppressing shared members is useful when you want to eliminate unnecessary duplication of data within the report.

### To suppress shared members:

At the point in the script where you want to suppress a shared member, type

**<SUPSHARE**

This suppresses the display of shared members for the duration of the report script. Use the <SUPSHAREOFF command to reset the display of shared members in the script.

See “Suppressing Page, Column, and Row Formatting” on page 36-14 for more information about suppressing data from your final report.

## Selecting Alias Names for Members

Aliases make reports easier to read and help your 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.

To Do This...	Use the Report Command
Display the alias set in the database outline's current alias table, without the member name.	OUTALT
Display the alias set in the database outline's current alias table, followed by the member name.	OUTALTMBR
Display the member name, followed by the alias set in the database outline's current alias table.	OUTMBRALT
Display the alias set in the database outline's current alias table, without the member name, for all members in the report.	OUTALTNAMES
Reset the default display of member names after using the OUTALTNAMES command.	OUTMBRNAMES
Include several alias tables within one report script.	OUTALTSELECT

For a complete listing of alias commands, see the online *Technical Reference* in the DOCS directory.

### To Display a Member Name and Alias

You can display members in a report as a combination of the member name and its alias. This lets you display more descriptive page, column, and row names, such as Diet Cola 100-20 or 100-30 Caffeine Free Cola.

To display a member name and alias, use the <OUTALTNAMES and <OUTALTMBR commands together in a report script, as shown in the following example:

```
<PAGE (Product, Measures)
<COLUMN (Scenario, Year)
{OUTALTNAMES}
<OUTALTMBR
Actual
<ICHILDREN Qtr1
<ROW (Market)
<IDESCENDANTS "300"
!
```

The report script produces the following 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 your 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 either 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 any final sorting of rows during formatting. Be careful when you place a sort command in your 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.

To Do This...	Use the Report Command
Sort all members alphabetically by the alias name of the member, if aliases are used in the report script.	SORTALTNAMES
Sort all following members in ascending order starting with the lowest generation and moving toward the highest generation.	SORTASC
Sort all following members in descending order starting with the highest generation and moving toward the lowest generation.	SORTDESC
Sort all following members according to the generation of the member in the database outline.	SORTGEN
Sort all following members according to the level of the member in the database outline.	SORTLEVEL

To Do This...	Use the Report Command
Sort all 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 online *Technical Reference* in the DOCS directory.

## Restricting and Ordering Data Values

Several Report Writer commands let you perform conditional retrieval and data sorting in your reports.

To Do This...	Use the Report Command
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
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

For the syntax and definitions of these commands, see the online *Technical Reference* in the DOCS directory. For detailed examples using these commands, see Chapter 37, “Examples of Report Scripts.”

Configurable variables are used during conditional retrievals. For information about setting the Report Writer configurable variables, see Chapter 38, “Optimizing Your Reports.”

## Defining 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, IDESC, or ICHILD).

Hyperion 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 and/or BOTTOM.
3. Applies ORDERBY.

Hyperion Essbase then returns rows and displays output.

See Chapter 35, “Quick Start to Report Scripts” for more information.

## Using TOP, BOTTOM, and ORDERBY with Sorting Commands

<TOP, <BOTTOM, and <ORDERBY commands sort a report output by its data values. Hyperion 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, Hyperion Essbase applies the sort command first, followed by <TOP and <BOTTOM, and then <ORDERBY. <ORDERBY is the final sort that takes place.

## Restricting Data Ranges

The arguments of the <RESTRICT command let you specify qualifications for selecting rows. Hyperion 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.

Hyperion 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)...) )
```

Only one <RESTRICT is allowed 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))
```

## Ordering Data

The <ORDERBY command orders the output rows according to the data values in the specified columns. You can specify either ascending <ASC (the default) or descending <DESC. You can specify different sorting directions in different columns of the same report.

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

Follow these guidelines when using the <ORDERBY command in a report script:

- 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 “ICHILD column dimension, <column ..., column member”).

## Specifying Rows to Return

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:** Hyperion 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))
```

This command returns the following 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))
```

This command returns the following 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 is equal to the number of rows in the <TOP plus the number of rows in the <BOTTOM commands.

## How Other Report Configurations Affect Row Specifications

When using the <TOP and <BOTTOM commands, be aware that some other commands affect their operation. In particular, Hyperion Essbase treats the <SUPMISSING, <SUPZEROS, and <SUPEMPTYROWS options as restrictions and applies them to the extracted rows along with the <RESTRICT command restrictions. Hyperion 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. Hyperion 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.

Because of this, it is a good idea to place all general formatting commands before a <COLUMN command, or a command that expands into column members. This guarantees that the formatting is executed. However, you should 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 your 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 Chapter 43, “Designing and Building Currency Conversion Applications.”

For the syntax and definitions of Report Writer commands, see the online *Technical Reference* in the DOCS directory.

## Generating Reports Using the C, Visual Basic, and Grid APIs

Use the following table to determine the report API calls that you can make:

To Do This...	Use the C API Function	Use the Visual Basic API Function	Use the 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 online *API Reference* in the DOCS directory for syntax and descriptions of these API functions.

This chapter includes report scripts that demonstrate report procedures and formats that are most frequently required in business settings. If you examine the techniques in these scripts and the resulting output, you can adapt them for use in your own reports.

The samples use both the Demo Basic and Sample Basic databases provided with your Hyperion Essbase server. Each sample identifies the database used. The scripts for these examples are available in your `\ESSBASE\APP\DEMO\BASIC` directory or your `\ESSBASE\APP\SAMPLE\BASIC` directory.

The sample reports in this chapter demonstrate the following techniques:

- “Sample 1: Creating a Different Format for Each Page” on page 37-2
- “Sample 2: Handling Missing Values” on page 37-4
- “Sample 3: Nesting Columns” on page 37-6
- “Sample 4: Grouping Rows” on page 37-8
- “Sample 5: Reporting on Different Combinations of Data” on page 37-14
- “Sample 6: Formatting Different Combinations of Data” on page 37-17
- “Sample 7: Using Aliases” on page 37-20
- “Sample 8: Creating Custom Headings and % Characters” on page 37-23
- “Sample 9: Creating Custom Page Headings” on page 37-27
- “Sample 10: Using Formulas” on page 37-32

- “Sample 11: Placing Two Page Layouts on the Same Page” on page 37-35
- “Sample 12: Formatting for Data Export” on page 37-37
- “Sample 13: Creating Asymmetric Columns” on page 37-39
- “Sample 14: Calculating Columns” on page 37-40
- “Sample 15: Calculating Rows” on page 37-44
- “Sample 16: Sorting by Top or Bottom Data Values” on page 37-51
- “Sample 17: Restricting Rows” on page 37-54
- “Sample 18: Ordering Data Values” on page 37-55
- “Sample 19: Narrowing Member Selection Criteria” on page 37-57
- “Sample 20: Using Attributes in Member Selection” on page 37-58

For fundamental information about reports and report scripts, see Chapter 35, “Quick Start to Report Scripts.” For detailed information about using Report Writer commands to write reports and reports scripts, see Chapter 36, “Developing Report Scripts.” For the syntax and usage of each Report Writer command, see the online *Technical Reference* in the DOCS directory.

## Sample 1: Creating a Different Format for Each Page

This sample report contains data for Actual Sales. Each report page shows a different Product. The report lists products on the same page until the maximum page length is reached. To place each Product on a separate page, you must use the PAGEONDIMENSION format command, as shown in Sample 2.

Because none of the cities in South sell Stereo or Compact\_Disc, the data values indicate #MISSING. You can represent missing values by suppressing the row or substituting a replacement text string, such as N/A. See “Sample 2: Handling Missing Values” on page 37-4 for an example of substituting page breaks and labels for missing values.

Sales Actual Stereo				
	Qtr1	Qtr2	Qtr3	Qtr4
	=====	=====	=====	=====
East	7,839	7,933	7,673	10,044
West	11,633	11,191	11,299	14,018
South	#Missing	#Missing	#Missing	#Missing
Market	19,472	19,124	18,972	24,062

Sales Actual Compact_Disc				
	Qtr1	Qtr2	Qtr3	Qtr4
	=====	=====	=====	=====
East	10,293	9,702	9,965	11,792
West	14,321	14,016	14,328	17,247
South	#Missing	#Missing	#Missing	#Missing
Market	24,614	23,718	24,293	29,039

Sales Actual Audio				
	Qtr1	Qtr2	Qtr3	Qtr4
	=====	=====	=====	=====
East	18,132	17,635	17,638	21,836
West	25,954	25,207	25,627	31,265
South	#Missing	#Missing	#Missing	#Missing
Market	44,086	42,842	43,265	53,101

Use the following script to create Sample 1:

```
<PAGE (Accounts, Scenario, Product)
Sales
Actual
<IDESCENDANTS Audio

      <COLUMN (Year)
      <CHILDREN Year

<ROW(Market)
<ICHILDREN Market
!
```

The ! report output command is required to generate the report.

Because the IDESC selection command is used for Audio, the report selects all three members. Only a single member is selected from the other page dimensions, Sales and Actual. As a result, the script creates three report pages. They display as one long report page unless you use the PAGEONDIMENSION format command, as shown in Sample 2.

This report script, ACTSALES.REP, is available in your \ESSBASE\APP\DEMO\BASIC directory.

## Sample 2: Handling Missing Values

This report has the same layout and member selection as Sample 1, and shows you how to use page breaks and labels for missing values.

Sales Actual Stereo				
	Qtr1	Qtr2	Qtr3	Qtr4
	=====	=====	=====	=====
East	7,839	7,933	7,673	10,044
West	11,633	11,191	11,299	14,018
South	N/A	N/A	N/A	N/A
Market	19,472	19,124	18,972	24,062

Sales Actual Compact_Disc				
	Qtr1	Qtr2	Qtr3	Qtr4
	=====	=====	=====	=====
East	10,293	9,702	9,965	11,792
West	14,321	14,016	14,328	17,247
South	N/A	N/A	N/A	N/A
Market	24,614	23,718	24,293	29,039

	Sales Actual Audio			
	Qtr1	Qtr2	Qtr3	Qtr4
	=====	=====	=====	=====
East	18,132	17,635	17,638	21,836
West	25,954	25,207	25,627	31,265
South	N/A	N/A	N/A	N/A
Market	44,086	42,842	43,265	53,101

Use the following script to create Sample 2:

```
<PAGE (Accounts, Scenario, Product)
Sales
Actual
<IDESCENDANTS Product
{ PAGEONDIMENSION Product }
{ MISSINGTEXT "N/A" }

      <COLUMN (Year)
      <CHILDREN Year

<ROW(Market)
<ICHILDREN Market
!
```

The PAGEONDIMENSION format command creates a page break whenever a member from the specified dimension changes. Because the report selects eight Product members, this creates an eight-page report.

The MISSINGTEXT format command substitutes any strings enclosed within double quotes into the #MISSING string. To suppress missing values, use the SUPMISSINGROWS command.

You can also combine format commands within one set of braces:

```
{ PAGEONDIMENSION Product MISSINGTEXT "N/A" }
```

This report script, MISS\_LBL.REP, is available in your \ESSBASE\APP\DEMO\BASIC directory.

## Sample 3: Nesting Columns

Each page produced by this report sample contains Sales information for a given Market. The report has two groups of columns across the page. The Actual and Budget members are the nested column group below Year members.

Note that the Actual and Budget members are on the same line in the report. You can put multiple commands on one line, but report commands are easier to read if they are spread out.

Sales East								
	Jan		Feb		Mar		Qtr1	
	Actual	Budget	Actual	Budget	Actual	Budget	Actual	Budget
	=====	=====	=====	=====	=====	=====	=====	=====
Stereo	2,788	2,950	2,482	2,700	2,569	2,700	7,839	8,350
Compact_Disc	3,550	3,450	3,285	3,250	3,458	3,250	10,293	9,950
Audio	6,338	6,400	5,767	5,950	6,027	5,950	18,132	18,300
Television	5,244	4,800	4,200	4,300	3,960	4,300	13,404	13,400
VCR	4,311	4,200	3,734	3,700	3,676	3,700	11,721	11,600
Camera	2,656	2,850	2,525	2,670	2,541	2,670	7,722	8,190
Visual	12,211	11,850	10,45	10,670	10,177	10,670	32,847	33,190
Product	18,549	18,250	16,226	16,620	16,204	16,620	50,979	51,490

Sales West								
	Jan		Feb		Mar		Qtr1	
	Actual	Budget	Actual	Budget	Actual	Budget	Actual	Budget
	=====	=====	=====	=====	=====	=====	=====	=====
Stereo	4,102	4,000	3,723	3,600	3,808	3,600	11,633	11,200
Compact_Disc	4,886	4,700	4,647	4,400	4,788	4,400	14,321	13,500
Audio	8,988	8,700	8,370	8,000	8,596	8,000	25,954	24,700
Television	5,206	5,100	4,640	4,600	4,783	4,600	14,629	14,300
VCR	4,670	4,650	4,667	4,200	4,517	4,200	13,854	13,050
Camera	3,815	4,050	3,463	3,750	3,478	3,750	10,756	11,550
Visual	13,691	13,800	12,770	12,550	12,778	12,550	39,239	38,900
Product	22,679	22,500	21,140	20,550	21,374	20,550	65,193	63,600

Sales South								
	Jan		Feb		Mar		Qtr1	
	Actual	Budget	Actual	Budget	Actual	Budget	Actual	Budget
	=====	=====	=====	=====	=====	=====	=====	=====
Television	3,137	3,400	2,929	3,100	2,815	3,100	8,881	9,600
VCR	3,225	3,400	3,206	3,100	3,120	3,100	9,551	9,600
Camera	2,306	2,400	2,167	2,400	2,168	2,400	6,641	7,200
Visual	8,668	9,200	8,302	8,600	8,103	8,600	25,073	26,400
Product	8,668	9,200	8,302	8,600	8,103	8,600	25,073	26,400

Sales Market								
	Jan		Feb		Mar		Qtr1	
	Actual	Budget	Actual	Budget	Actual	Budget	Actual	Budget
	=====	=====	=====	=====	=====	=====	=====	=====
Stereo	6,890	6,950	6,205	6,300	6,377	6,300	19,472	19,550
Compact_Disc	8,436	8,150	7,932	7,650	8,246	7,650	24,614	23,450
Audio	15,326	15,100	14,137	13,950	14,623	13,950	44,086	43,000
Television	13,587	13,300	11,769	12,000	11,558	12,000	36,914	37,300
VCR	12,206	12,250	11,607	11,000	11,313	11,000	35,126	34,250
Camera	8,777	9,300	8,155	8,820	8,187	8,820	25,119	26,940
Visual	34,570	34,850	31,531	31,820	31,058	31,820	97,159	98,490
Product	49,896	49,950	45,668	45,770	45,681	45,770	141,245	141,490

Use the following script to create Sample 3:

```

<PAGE (Accounts, Market)
Sales
<ICHILDREN Market
{ PAGEONDIMENSION Market }
{ SUPMISSINGROWS }

      <COLUMN (Year, Scenario)
      <ICHILDREN Qtr1
      Actual Budget

<ROW(Product)
<IDESCENDANTS Product
!
```

The report selects four Markets because the <ICHILDREN command is applied to Market. Only Sales is selected from the other page dimension, so the report has four pages.

For the South, all the rows of Product data are not displayed. Recall that the cities in the South do not sell every Product. The report uses the SUPMISSINGROWS format command to suppress the output of any member rows with all missing values.

This report script, COLGROUP.REP, is available in your \ESSBASE\APP\DEMO\BASIC directory.

## Sample 4: Grouping Rows

Each page of this report contains Sales information for a given Market. The report page contains members for both Product and Year as groups of rows down the page. This script creates a four-page report because the page dimensions and their member selections are the same as in Sample 3. The row/column layout is switched because the row and column dimensions are different.

		Sales East		
		Actual	Budget	Variance
		=====	=====	=====
Stereo	Qtr1	7,839	8,350	(511)
	Qtr2	7,933	8,150	(217)
	Qtr3	7,673	8,350	(677)
	Qtr4	10,044	10,400	(356)
	Year	33,489	35,250	(1,761)
Compact_Disc	Qtr1	10,293	9,950	343
	Qtr2	9,702	9,750	(48)
	Qtr3	9,965	10,050	(85)
	Qtr4	11,792	12,550	(758)
	Year	41,752	42,300	(548)
Audio	Qtr1	18,132	18,300	(168)
	Qtr2	17,635	17,900	(265)
	Qtr3	17,638	18,400	(762)
	Qtr4	21,836	22,950	(1,114)
	Year	75,241	77,550	(2,309)
Television	Qtr1	13,404	13,400	4
	Qtr2	12,115	12,900	(785)
	Qtr3	15,014	14,200	814
	Qtr4	17,861	17,300	561
	Year	58,394	57,800	594
VCR	Qtr1	11,721	11,600	121
	Qtr2	10,999	11,100	(101)
	Qtr3	13,217	11,800	1,417
	Qtr4	14,386	14,900	(514)
	Year	50,323	49,400	923
Camera	Qtr1	7,722	8,190	(468)
	Qtr2	7,581	8,210	(629)
	Qtr3	8,181	8,630	(449)
	Qtr4	10,853	11,550	(697)
	Year	34,337	36,580	(2,243)
Visual	Qtr1	32,847	33,190	(343)
	Qtr2	30,695	32,210	(1,515)
	Qtr3	36,412	34,630	1,782
	Qtr4	43,100	43,750	(650)
	Year	143,054	143,780	(726)
Product	Qtr1	50,979	51,490	(511)
	Qtr2	48,330	50,110	(1,780)
	Qtr3	54,050	53,030	1,020
	Qtr4	64,936	66,700	(1,764)
	Year	218,295	221,330	(3,035)

		Sales West		
		Actual	Budget	Variance
		=====	=====	=====
Stereo	Qtr1	11,633	11,200	433
	Qtr2	11,191	11,050	141
	Qtr3	11,299	11,650	(351)
	Qtr4	14,018	14,500	(482)
	Year	48,141	48,400	(259)
Compact_Disc	Qtr1	14,321	13,500	821
	Qtr2	14,016	13,500	516
	Qtr3	14,328	14,300	28
	Qtr4	17,247	16,700	547
	Year	59,912	58,000	1,912
Audio	Qtr1	25,954	24,700	1,254
	Qtr2	25,207	24,550	657
	Qtr3	25,627	25,950	(323)
	Qtr4	31,265	31,200	65
	Year	108,053	106,400	1,653
Television	Qtr1	14,629	14,300	329
	Qtr2	14,486	13,800	686
	Qtr3	14,580	14,000	580
	Qtr4	20,814	19,400	1,414
	Year	64,509	61,500	3,009
VCR	Qtr1	13,854	13,050	804
	Qtr2	13,156	12,600	556
	Qtr3	15,030	13,750	1,280
	Qtr4	18,723	17,950	773
	Year	60,763	57,350	3,413
Camera	Qtr1	10,756	11,550	(794)
	Qtr2	10,573	11,400	(827)
	Qtr3	10,735	11,550	(815)
	Qtr4	13,906	15,000	(1,094)
	Year	45,970	49,500	(3,530)
Visual	Qtr1	39,239	38,900	339
	Qtr2	38,215	37,800	415
	Qtr3	40,345	39,300	1,045
	Qtr4	53,443	52,350	1,093
	Year	171,242	168,350	2,892
Product	Qtr1	65,193	63,600	1,593
	Qtr2	63,422	62,350	1,072
	Qtr3	65,972	65,250	722
	Qtr4	84,708	83,550	1,158
	Year	279,295	274,750	4,545

		Sales South		
		Actual	Budget	Variance
		=====	=====	=====
Television	Qtr1	8,881	9,600	(719)
	Qtr2	8,627	9,300	(673)
	Qtr3	8,674	9,300	(626)
	Qtr4	12,919	12,600	319
	Year	39,101	40,800	(1,699)
VCR	Qtr1	9,551	9,600	(49)
	Qtr2	9,049	9,300	(251)
	Qtr3	9,998	10,000	(2)
	Qtr4	12,923	13,600	(677)
	Year	41,521	42,500	(979)
Camera	Qtr1	6,641	7,200	(559)
	Qtr2	6,765	7,350	(585)
	Qtr3	6,798	7,500	(702)
	Qtr4	9,486	10,200	(714)
	Year	29,690	32,250	(2,560)
Visual	Qtr1	25,073	26,400	(1,327)
	Qtr2	24,441	25,950	(1,509)
	Qtr3	25,470	26,800	(1,330)
	Qtr4	35,328	36,400	(1,072)
	Year	110,312	115,550	(5,238)
Product	Qtr1	25,073	26,400	(1,327)
	Qtr2	24,441	25,950	(1,509)
	Qtr3	25,470	26,800	(1,330)
	Qtr4	35,328	36,400	(1,072)
	Year	110,312	115,550	(5,238)

		Sales Market		
		Actual	Budget	Variance
		=====	=====	=====
Stereo	Qtr1	19,472	19,550	(78)
	Qtr2	19,124	19,200	(76)
	Qtr3	18,972	20,000	(1,028)
	Qtr4	24,062	24,900	(838)
	Year	81,630	83,650	(2,020)
Compact_Disc	Qtr1	24,614	23,450	1,164
	Qtr2	23,718	23,250	468
	Qtr3	24,293	24,350	(57)
	Qtr4	29,039	29,250	(211)
	Year	101,664	100,300	1,364
Audio	Qtr1	44,086	43,000	1,086
	Qtr2	42,842	42,450	392
	Qtr3	43,265	44,350	(1,085)
	Qtr4	53,101	54,150	(1,049)
	Year	183,294	183,950	(656)
Television	Qtr1	36,914	37,300	(386)
	Qtr2	35,228	36,000	(772)
	Qtr3	38,268	37,500	768
	Qtr4	51,594	49,300	2,294
	Year	162,004	160,100	1,904
VCR	Qtr1	35,126	34,250	876
	Qtr2	33,204	33,000	204
	Qtr3	38,245	35,550	2,695
	Qtr4	46,032	46,450	(418)
	Year	152,607	149,250	3,357
Camera	Qtr1	25,119	26,940	(1,821)
	Qtr2	24,919	26,960	(2,041)
	Qtr3	25,714	27,680	(1,966)
	Qtr4	34,245	36,750	(2,505)
	Year	109,997	118,330	(8,333)
Visual	Qtr1	97,159	98,490	(1,331)
	Qtr2	93,351	95,960	(2,609)
	Qtr3	102,227	100,730	1,497
	Qtr4	131,871	132,500	(629)
	Year	424,608	427,680	(3,072)
Product	Qtr1	141,245	141,490	(245)
	Qtr2	136,193	138,410	(2,217)
	Qtr3	145,492	145,080	412
	Qtr4	184,972	186,650	(1,678)
	Year	607,902	611,630	(3,728)

Use the following script to create Sample 4:

```
<PAGE (Accounts, Market)
Sales
<CHILDREN Market
{ PAGEONDIMENSION Market }
{ SUPMISSINGROWS }

        <COLUMN (Scenario)
        <CHILDREN Scenario

<ROW(Product3, Year)
<CHILDREN Year
<IDESCENDANTS Product
!
```

This report script, ROWGROUP.REP, is available in your \ESSBASE\APP\DEMO\BASIC directory.

## Sample 5: Reporting on Different Combinations of Data

Each page represents a different combination of Product, Market, and Budget data. The total number of pages is determined by the number of Market and Product members. This section shows a representative part of the output.

Some data values have four decimal places. The number of decimal places, by default, is output to the true number of decimal values of the data cell. “Sample 6: Formatting Different Combinations of Data” on page 37-17 uses the DECIMAL format command to define a specific number of places.

The member selection commands select three Product members and fourteen Market members. This produces a 42-page report. The number of report pages is determined by multiplying the number of members selected from each page dimension.

	Budget Audio New_York				
	Qtr1	Qtr2	Qtr3	Qtr4	Year
	=====	=====	=====	=====	=====
Sales	6,400	6,400	6,700	8,350	27,850
Cost_of_Goods_Sold	3,012	3,012	3,146	3,973	13,143
Margin	3,388	3,388	3,554	4,377	14,707
Marketing	525	515	475	555	2,070
Payroll	1,950	1,950	1,950	1,950	7,800
Miscellaneous	0	0	0	0	0
Total_Expenses	2,475	2,465	2,425	2,505	9,870
Profit	913	923	1,129	1,872	4,837
Profit_%	14	14	17	22	17
Margin_%	53	53	53	52	53

Budget Audio Boston					
	Qtr1	Qtr2	Qtr3	Qtr4	Year
	=====	=====	=====	=====	=====
Sales	6,050	5,750	5,900	7,350	25,050
Cost_of_Goods_Sold	2,829	2,695	2,762	3,413	11,699
Margin	3,221	3,055	3,138	3,937	13,351
Marketing	410	400	400	520	1,730
Payroll	1,590	1,590	1,590	1,590	6,360
Miscellaneous	0	0	0	0	0
Total_Expenses	2,000	1,990	1,990	2,110	8,090
Profit	1,221	1,065	1,148	1,827	5,261
Profit_%	20	19	19	25	21
Margin_%	53	53	53	54	53

Budget Product Market					
	Qtr1	Qtr2	Qtr3	Qtr4	Year
	=====	=====	=====	=====	=====
Sales	141,490	138,410	145,080	186,650	611,630
Cost_of_Goods_Sold	55,860	54,579	57,379	73,276	241,093
Margin	85,630	83,831	87,702	113,374	370,537
Marketing	10,555	10,680	10,780	13,915	45,930
Payroll	43,234	43,248	43,248	43,248	172,978
Miscellaneous	0	0	0	0	0
Total_Expenses	53,789	53,928	54,028	57,163	218,908
Profit	31,841	29,903	33,674	56,211	151,629
Profit_%	23	22	23	30	25
Margin_%	61	61	60	61	61

Use the following script to create Sample 5:

```
<PAGE (Scenario, Product, Market)
Budget
<ICHILDREN Product
<IDESCENDANTS Market
{ PAGEONDIMENSION Product } // New page at each new Product
{ PAGEONDIMENSION Market } // New page at each new Market
    <COLUMN (Year)
    <ICHILDREN Year

<ROW(Accounts)
<DESCENDANTS Accounts
    !
```

This report script, COMBO1.REP, is available in your \ESSBASE\APP\DEMO\BASIC directory.

## Sample 6: Formatting Different Combinations of Data

This report uses the same layout and member selection as Sample 5, and adds more formatting in the report body. Note the use of line formatting.

	Budget Audio New_York				
	Qtr1	Qtr2	Qtr3	Qtr4	Year
	=====	=====	=====	=====	=====
Sales	6,400	6,400	6,700	8,350	27,850
Cost_of_Goods_Sold	3,012	3,012	3,146	3,973	13,143
	-----	-----	-----	-----	-----
Margin	3,388	3,388	3,554	4,377	14,707
Marketing	525	515	475	555	2,070
Payroll	1,950	1,950	1,950	1,950	7,800
Miscellaneous	0	0	0	0	0
	-----	-----	-----	-----	-----
Total_Expenses	2,475	2,465	2,425	2,505	9,870
Profit	913	923	1,129	1,872	4,837
	=====	=====	=====	=====	=====
Profit_%	14.27	14.42	16.85	22.42	17.37
Margin_%	52.94	52.94	53.04	52.42	52.81

Budget Audio Boston					
	Qtr1	Qtr2	Qtr3	Qtr4	Year
	=====	=====	=====	=====	=====
Sales	6,050	5,750	5,900	7,350	25,050
Cost_of_Goods_Sold	2,829	2,695	2,762	3,413	11,699
	-----	-----	-----	-----	-----
Margin	3,221	3,055	3,138	3,937	13,351
Marketing	410	400	400	520	1,730
Payroll	1,590	1,590	1,590	1,590	6,360
Miscellaneous	0	0	0	0	0
	-----	-----	-----	-----	-----
Total_Expenses	2,000	1,990	1,990	2,110	8,090
Profit	1,221	1,065	1,148	1,827	5,261
	=====	=====	=====	=====	=====
Profit_%	20.18	18.52	19.46	24.86	21.00
Margin_%	53.24	53.13	53.19	53.56	53.30

Use the following script to create Sample 6:

```

<PAGE (Scenario, Product, Market)
{ PAGEONDIMENSION Product PAGEONDIMENSION Market }
Budget
<ICHILDREN Product
<IDESCENDANTS Market

      <COLUMN (Year)
      <ICHILDREN Year

<ROW(Accounts)

{ SUPBRACKETS DECIMAL 0 }
Sales
Cost_of_Goods_Sold

{ UDATA "-" } //line formatting command
Margin
    
```

```

{ SKIP }
Marketing
Payroll
Miscellaneous

{ UDATA "-" }          //line formatting command
Total_Expenses

{ SKIP }
Profit

{ UDATA DECIMAL 2 }    //line formatting command
Profit_%
Margin_%
!
```

Format commands apply to members that follow the commands. The report begins each new page with the formats in place at the end of the previous report page. For example, if a report page ends with two decimal places, the following page begins with two decimal places. This report demonstrates the use of several important format commands:

- **DECIMAL**—The script for this report specifies the DECIMAL 0 format command before the Sales member.
- **SUPBRACKETS**—By default, negative numbers are enclosed in brackets, (). The SUPBRACKETS format command causes negative numbers to be output with a minus sign.
- **UDATA**—The UDATA command places underline characters under data columns. The character is specified within double quotes. The default is a double underline.

This report script, COMBO2.REP, is available in your \ESSBASE\APP\DEMO\BASIC directory.

## Sample 7: Using Aliases

This report outputs members in the middle of a page and uses aliases or alternate names. The default row member indentation is turned off.

Stereo Market				
Qtr4			Year	
Actual	Budget		Actual	Budget
=====	=====		=====	=====
24,062	24,900	Sales	81,630	83,650
13,937	14,442	COGS	47,654	48,517
-----				
10,125	10,458	Margin	33,976	35,133
1,438	1,600	Marketing	4,933	5,465
7,110	6,840	Payroll	28,440	27,360
-200	0	Misc.	-143	0
-----				
8,348	8,440	Total_Expenses	33,230	32,825
1,777	2,018	Profit	746	2,308
=====	=====		=====	=====
7.39	8.10	Profit_%	0.91	2.76
42.08	42.00	Margin_%	41.62	42.00

Compact_Disc Market				
Qtr4			Period	
Actual	Budget		Actual	Budget
=====	=====		=====	=====
29,039	29,250	Sales	101,664	100,300
10,830	11,115	COGS	38,120	38,114
-----	-----		-----	-----
18,209	18,135	Margin	63,544	62,186
1,669	1,780	Marketing	6,067	5,975
5,721	5,415	Payroll	22,200	21,660
-226	0	Misc.	97	0
-----	-----		-----	-----
7,164	7,195	Total_Expenses	28,364	27,635
11,045	10,940	Profit	35,180	34,551
=====	=====		=====	=====
38.04	37.40	Profit_%	34.60	34.45
62.71	62.00	Margin_%	62.50	62.00

Use the following script to create Sample 7:

```

<PAGE (Product, Market)
{ PAGEONDIMENSION Product }
{ PAGEONDIMENSION Market }
<IDESCENDANTS Product
{ DECIMAL 0 }
<SYM

      <COLUMN (Year, Scenario)
      Qtr4 Year
      Actual Budget
<ROW(Accounts)
{ SUPBRACKETS OUTALTNAMES NOINDENTGEN ORDER 1,2,0,3,4 }
Sales Cost_of_Goods_Sold
{ UDATA "-" }
Margin
{ SKIP }
Marketing Payroll Miscellaneous
{ UDATA "-" }
Total_Expenses
{ SKIP }
Profit
{ UDATA DECIMAL 2 }
Profit_%
Margin_%
!
```

The `SYM` command forces the report to output symmetric column groups. The default is to display two columns: one for Qtr4 Actual and one for Year Budget. Because the report calls for Actual and Budget under both Qtr4 and Year, the `SYM` command is required. Alternatively, repeat the Actual and Budget names under Qtr4 and Year.

The `OUTALTNames` format command causes the report to use aliases or alternate names instead of member names.

The `NOINDENTGEN` format command causes row members to not be indented. By default, members are indented two spaces for each level.

The `ORDER` command moves specified output columns to new locations. The row name is considered column 0.

The `FIXCOLUMNS` format command restricts the number of output columns. Reports often require both `ORDER` and `FIXCOLUMNS`. You can use `ORDER` to remove unwanted columns, and `FIXCOLUMNS` to stop these columns from displaying after the report columns.

This report script, `MIDDLE.REP`, is available in your `\ESSBASE\APP\DEMO\BASIC` directory.

## Sample 8: Creating Custom Headings and % Characters

This report displays custom headings and percent sign (%) characters after each data value. This section shows a representative part of the output.

Prepared by: Admin	The Electronics Club					Page: 1 09/21/95
	Profit_% Actual Stereo					
	Jan	Feb	Mar	Apr	May	Jun
	=====	=====	=====	=====	=====	=====
New_York	1.43%	-10.00%	-3.51%	-2.22%	1.14%	-6.18%
Boston	-0.34%	-2.51%	-4.44%	-4.89%	-7.02%	-13.15%
Chicago	-0.65%	-0.72%	-2.28%	-3.53%	-6.33%	-10.79%
East	0.18%	-4.47%	-3.39%	-3.41%	-3.60%	-9.70%
San_Francisco	1.43%	-1.87%	4.42%	2.15%	-1.26%	0.66%
Seattle	0.95%	-5.66%	1.42%	-6.82%	-11.47%	-12.34%
Denver	3.03%	-1.11%	-5.88%	-6.52%	-5.17%	-13.83%
Los_Angeles	-1.50%	-3.94%	-2.86%	-3.29%	3.12%	-2.51%
West	0.98%	-2.95%	-0.13%	-2.81%	-2.62%	-5.61%
Dallas	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Houston	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Phoenix	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
South	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Market	0.65%	-3.56%	-1.44%	-3.06%	-3.03%	-7.29%

Prepared by: Admin	The Electronics Club					Page: 2 09/21/95
	Profit_% Actual Compact_Disc					
	Jan	Feb	Mar	Apr	May	Jun
	=====	=====	=====	=====	=====	=====
New_York	32.51%	29.95%	35.30%	32.70%	30.45%	31.73%
Boston	33.42%	27.92%	33.98%	30.74%	27.45%	30.85%
Chicago	34.29%	30.48%	26.33%	28.83%	28.11%	33.76%
East	33.35%	29.50%	32.30%	30.92%	28.77%	32.09%
San_Francisco	37.77%	35.02%	33.41%	33.23%	35.32%	37.95%
Seattle	40.41%	38.33%	38.89%	37.06%	37.01%	38.29%
Denver	31.93%	32.10%	34.82%	29.15%	32.71%	30.85%
Los_Angeles	31.65%	30.22%	30.22%	31.45%	27.06%	33.20%
West	35.51%	33.94%	34.21%	32.77%	33.16%	35.25%
Dallas	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Houston	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Phoenix	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
South	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Market	34.60%	32.10%	33.41%	32.01%	31.35%	33.97%

Prepared by: Admin	The Electronics Club					Page: 3 09/21/95
	Profit_% Actual Audio					
	Jan	Feb	Mar	Apr	May	Jun
	=====	=====	=====	=====	=====	=====
New_York	19.35%	13.64%	18.64%	16.55%	16.70%	14.65%
Boston	18.34%	14.44%	18.94%	14.94%	12.14%	12.42%
Chicago	18.50%	16.67%	13.18%	14.12%	12.70%	13.74%
East	18.76%	14.88%	17.09%	15.32%	14.05%	13.68%
San_Francisco	20.32%	17.38%	18.92%	18.03%	18.23%	20.57%
Seattle	23.36%	21.40%	23.37%	20.17%	18.82%	19.04%
Denver	18.36%	17.25%	18.88%	13.43%	15.84%	12.14%
Los_Angeles	17.15%	14.76%	15.44%	15.76%	15.10%	17.07%
West	19.75%	17.53%	19.00%	16.88%	17.01%	17.52%
Dallas	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Houston	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Phoenix	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
South	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Market	19.34%	16.45%	18.21%	16.24%	15.78%	15.96%

Prepared by: Admin		The Electronics Club				Page: 8 09/21/95	
Profit_% Actual Product							
	Jan	Feb	Mar	Apr	May	Jun	
	=====	=====	=====	=====	=====	=====	
New_York	22.71%	21.43%	13.11%	10.54%	9.73%	13.16%	
Boston	24.98%	23.25%	19.95%	18.00%	17.03%	18.62%	
Chicago	22.01%	17.94%	18.14%	15.45%	18.70%	16.01%	
East	23.19%	20.84%	16.89%	14.42%	14.94%	15.78%	
San_Francisco	23.71%	20.60%	21.93%	20.45%	21.44%	19.98%	
Seattle	21.06%	21.05%	21.24%	19.00%	21.72%	15.13%	
Denver	21.61%	16.01%	19.79%	14.81%	20.66%	13.89%	
Los_Angeles	17.54%	15.51%	17.03%	14.33%	17.59%	16.09%	
West	21.02%	18.35%	19.99%	17.26%	20.30%	16.61%	
Dallas	15.67%	16.50%	15.32%	13.93%	20.36%	15.49%	
Houston	20.01%	20.29%	20.62%	15.87%	23.60%	12.38%	
Phoenix	20.01%	16.12%	17.18%	16.50%	21.39%	15.22%	
South	18.39%	17.53%	17.59%	15.36%	21.66%	14.46%	
Market	21.37%	19.09%	18.46%	15.92%	18.67%	15.93%	

Use the following script to create Sample 8:

```

<PAGE (Accounts, Scenario, Product)
{ PAGEONDIMENSION Product } // New page when Product changes
Profit_%
Actual
<IDESCENDANTS Product

      <COLUMN (Year)
      Jan Feb Mar Apr May Jun

<ROW(Market)

{ STARTHEADING
TEXT 1 "Prepared by:"
    14 "*USERNAME"
    C "The Electronics Club"
    65 "*PAGESTRING"
TEXT 65 "*DATE"
SKIP
ENDHEADING }

```

```
{ Decimal 2 AFTER "%" SUPBRACKETS } // Place % at end and  
// suppress bracket  
<IDESCENDANTS Market  
!
```

Each data value in the report has a percent sign, %. This label is defined with the AFTER “%” format command. You can specify any character within quotation marks.

This report has custom headings at the top of each page. All format commands specified between the STARTHEADING and ENDHEADING format commands are displayed at the top of each report page.

TEXT format commands define text labels. The report generator provides *dynamic* text with *\*options*. See the online *Technical Reference* in the DOCS directory for a full list of the available options. This report uses the following options:

- \*USERNAME—which outputs the user name used when connecting to Hyperion Essbase
- \*PAGESTRING—which outputs the current page number of the report
- C—which centers the report title

This report script, HEADING1.REP, is available in your \ESSBASE\APP\DEMO\BASIC directory.

## Sample 9: Creating Custom Page Headings

This report builds on Sample 8 by adding custom page headings. By default, page dimension members are output at the top center of a report page.

Prepared by :admin	The Electronics Club					Page: 1
	Actual Profit by Product					12/12/95
Product: Stereo						
	Jan	Feb	Mar	Apr	May	Jun
New York	1.43%	-10.00%	-3.51%	-2.22%	1.14%	-6.18%
Boston	-0.34%	-2.51%	-4.44%	-4.89%	-7.02%	-13.15%
Chicago	-0.65%	-0.72%	-2.28%	-3.53%	-6.33%	-10.79%
San Francisco	1.43%	-1.87%	4.42%	2.15%	-1.26%	0.66%
Seattle	0.95%	-5.66%	1.42%	-6.82%	-11.47%	-12.34%
Denver	3.03%	-1.11%	-5.88%	-6.52%	-5.17%	-13.83%
Los Angeles	-1.50%	-3.94%	-2.86%	-3.29%	3.12%	-2.51%
Dallas	#Missing	#Missing	#Missing	#Missing	#Missing	#Missing
Houston	#Missing	#Missing	#Missing	#Missing	#Missing	#Missing
Phoenix	#Missing	#Missing	#Missing	#Missing	#Missing	#Missing
East	0.18%	-4.47%	-3.39%	-3.41%	-3.60%	-9.70%
West	0.98%	-2.95%	-0.13%	-2.81%	-2.62%	-5.61%
South	#Missing	#Missing	#Missing	#Missing	#Missing	#Missing
Market	0.65%	-3.56%	-1.44%	-3.06%	-3.03%	-7.29%

Prepared by :admin	The Electronics Club					Page: 2
	Actual Profit by Product					12/12/95
Pageduct:Compact Disc						
	Jan	Feb	Mar	Apr	May	Jun
New York	32.51%	29.95%	35.30%	32.70%	30.45%	31.73%
Boston	33.42%	27.92%	33.98%	30.74%	27.45%	30.85%
Chicago	34.29%	30.48%	26.33%	28.83%	28.11%	33.76%
San Francisco	37.77%	35.02%	33.41%	33.23%	35.32%	37.95%
Seattle	40.41%	38.33%	38.89%	37.06%	37.01%	38.29%
Denver	31.93%	32.10%	34.82%	29.15%	32.71%	30.85%
Los Angeles	31.65%	30.22%	30.22%	31.45%	27.06%	33.20%
Dallas	#Missing	#Missing	#Missing	#Missing	#Missing	#Missing
Houston	#Missing	#Missing	#Missing	#Missing	#Missing	#Missing
Phoenix	#Missing	#Missing	#Missing	#Missing	#Missing	#Missing
East	33.35%	29.50%	32.30%	30.92%	28.77%	32.09%
West	35.51%	33.94%	34.21%	32.77%	33.16%	35.25%
South	#Missing	#Missing	#Missing	#Missing	#Missing	#Missing
Market	34.60%	32.10%	33.41%	32.01%	31.35%	33.97%

Examples of Report Scripts

Prepared by :admin		The Electronics Club					Page: 3
		Actual Profit by Product					12/12/95
Product: Audio							
	Jan	Feb	Mar	Apr	May	Jun	
New York	19.35%	13.64%	18.64%	16.55%	16.70%	14.65%	
Boston	18.34%	14.44%	18.94%	14.94%	12.14%	12.42%	
Chicago	18.50%	16.67%	13.18%	14.12%	12.70%	13.74%	
San Francisco	20.32%	17.38%	18.92%	18.03%	18.23%	20.57%	
Seattle	23.36%	21.40%	23.37%	20.17%	18.82%	19.04%	
Denver	18.36%	17.25%	18.88%	13.43%	15.84%	12.14%	
Los Angeles	17.15%	14.76%	15.44%	15.76%	15.10%	17.07%	
Dallas	#Missing	#Missing	#Missing	#Missing	#Missing	#Missing	
Houston	#Missing	#Missing	#Missing	#Missing	#Missing	#Missing	
Phoenix	#Missing	#Missing	#Missing	#Missing	#Missing	#Missing	
East	18.76%	14.88%	17.09%	15.32%	14.05%	13.68%	
West	19.75%	17.53%	19.00%	16.88%	17.01%	17.52%	
South	#Missing	#Missing	#Missing	#Missing	#Missing	#Missing	
Market	19.34%	16.45%	18.21%	16.24%	15.78%	15.96%	

Prepared by :admin		The Electronics Club					Page: 4
		Actual Profit by Product					12/12/95
Product: Television							
	Jan	Feb	Mar	Apr	May	Jun	
New York	19.45%	28.80%	3.29%	-4.14%	-1.98%	7.73%	
Boston	34.53%	40.00%	24.92%	19.20%	15.00%	22.31%	
Chicago	20.61%	12.86%	15.29%	7.36%	19.23%	10.29%	
San Francisco	16.53%	12.59%	17.52%	10.74%	18.70%	7.43%	
Seattle	9.22%	12.29%	14.89%	18.39%	22.94%	7.87%	
Denver	17.69%	0.24%	16.39%	10.75%	21.05%	12.36%	
Los Angeles	10.41%	6.25%	10.48%	3.00%	14.52%	5.34%	
Dallas	2.24%	6.23%	5.89%	5.38%	15.48%	4.75%	
Houston	11.04%	8.65%	11.52%	0.54%	14.38%	2.19%	
Phoenix	11.09%	5.81%	8.58%	7.82%	12.47%	10.94%	
East	24.69%	26.95%	13.96%	6.89%	10.43%	13.14%	
West	13.41%	8.23%	14.57%	10.31%	18.95%	7.91%	
South	7.55%	6.73%	8.38%	4.90%	14.19%	5.97%	
Market	16.41%	14.54%	12.86%	7.86%	14.93%	9.28%	

Prepared by :admin	The Electronics Club					Page: 5
	Actual Profit by Product					12/12/95
Product:VCR						
	Jan	Feb	Mar	Apr	May	Jun
New York	34.65%	32.38%	19.50%	20.96%	14.14%	21.57%
Boston	31.44%	28.43%	22.33%	29.27%	25.86%	27.89%
Chicago	31.97%	29.00%	30.95%	27.85%	28.81%	28.26%
San Francisco	32.28%	31.47%	29.50%	31.81%	30.89%	26.42%
Seattle	31.06%	33.42%	29.99%	26.21%	26.86%	16.40%
Denver	31.18%	29.86%	30.39%	27.87%	29.66%	19.05%
Los Angeles	29.14%	29.11%	28.16%	26.27%	27.31%	29.36%
Dallas	32.17%	31.32%	27.11%	27.78%	27.55%	29.86%
Houston	34.07%	35.53%	35.29%	31.06%	32.90%	20.63%
Phoenix	32.91%	29.86%	30.39%	27.87%	29.66%	19.05%
East	32.87%	29.99%	24.43%	25.95%	22.99%	25.79%
West	30.88%	30.94%	29.42%	28.19%	28.76%	23.61%
South	32.99%	32.22%	30.83%	28.84%	29.89%	23.71%
Market	32.14%	30.99%	28.19%	27.62%	27.19%	24.37%

Prepared by :admin	The Electronics Club					Page: 6
	Actual Profit by Product					12/12/95
Product:Camera						
	Jan	Feb	Mar	Apr	May	Jun
New York	16.35%	12.30%	6.72%	4.26%	4.66%	6.67%
Boston	10.55%	6.52%	10.66%	4.99%	17.87%	10.74%
Chicago	17.57%	12.68%	13.67%	12.95%	16.71%	13.10%
San Francisco	30.97%	25.36%	25.64%	24.43%	21.37%	25.96%
Seattle	16.69%	11.96%	12.33%	7.91%	20.53%	13.56%
Denver	22.36%	12.31%	12.19%	7.77%	20.53%	13.56%
Los Angeles	14.96%	13.19%	15.65%	11.15%	14.95%	11.30%
Dallas	11.68%	8.99%	11.41%	4.72%	17.62%	10.47%
Houston	12.07%	10.99%	9.83%	11.79%	21.17%	13.15%
Phoenix	14.36%	11.24%	10.13%	13.31%	22.56%	15.70%
East	15.06%	10.77%	10.07%	7.41%	12.67%	9.99%
West	22.31%	16.92%	17.63%	14.32%	19.47%	17.24%
South	12.71%	10.38%	10.47%	9.94%	20.45%	13.11%
Market	17.59%	13.28%	13.39%	11.04%	17.68%	13.92%

Examples of Report Scripts

Prepared by :admin	The Electronics Club					Page: 7
	Actual Profit by Product					12/12/95
Product:Visual						
	Jan	Feb	Mar	Apr	May	Jun
New York	24.40%	25.71%	9.65%	6.67%	5.23%	12.31%
Boston	28.51%	28.19%	20.58%	19.75%	19.82%	21.89%
Chicago	23.86%	18.63%	20.75%	16.17%	21.97%	17.24%
San Francisco	25.96%	22.72%	24.00%	22.11%	23.63%	19.59%
Seattle	19.41%	20.82%	19.77%	18.23%	23.54%	12.48%
Denver	23.89%	15.13%	20.45%	15.73%	23.82%	15.05%
Los Angeles	17.77%	15.97%	17.99%	13.38%	19.15%	15.46%
Dallas	15.67%	16.50%	15.32%	13.93%	20.36%	15.49%
Houston	20.01%	20.29%	20.62%	15.87%	23.60%	12.38%
Phoenix	20.01%	16.12%	17.18%	16.50%	21.39%	15.22%
East	25.49%	24.13%	16.77%	13.89%	15.46%	16.94%
West	21.85%	18.89%	20.65%	17.52%	22.43%	16.00%
South	18.39%	17.53%	17.59%	15.36%	21.66%	14.46%
Market	22.26%	20.27%	18.58%	15.76%	19.98%	15.91%

Prepared by :admin	The Electronics Club					Page: 8
	Actual Profit by Product					12/12/95
Product:Product						
	Jan	Feb	Mar	Apr	May	Jun
New York	22.71%	21.43%	13.11%	10.54%	9.73%	13.16%
Boston	24.98%	23.25%	19.95%	18.00%	17.03%	18.62%
Chicago	22.01%	17.94%	18.14%	15.45%	18.70%	16.01%
San Francisco	23.71%	20.60%	21.93%	20.45%	21.44%	19.98%
Seattle	21.06%	21.05%	21.24%	19.00%	21.72%	15.13%
Denver	21.61%	16.01%	19.79%	14.81%	20.66%	13.89%
Los Angeles	17.54%	15.51%	17.03%	14.33%	17.59%	16.09%
Dallas	15.67%	16.50%	15.32%	13.93%	20.36%	15.49%
Houston	20.01%	20.29%	20.62%	15.87%	23.60%	12.38%
Phoenix	20.01%	16.12%	17.18%	16.50%	21.39%	15.22%
East	23.19%	20.84%	16.89%	14.42%	14.94%	15.78%
West	21.02%	18.35%	19.99%	17.26%	20.30%	16.61%
South	18.39%	17.53%	17.59%	15.36%	21.66%	14.46%
Market	21.37%	19.09%	18.46%	15.92%	18.67%	15.93%

Use the following script to create Sample 9:

```

<PAGE (Accounts, Scenario, Product)
<IDESCENDANTS Product
<SORTLEVEL
{ PAGEONDIMENSION Product }
{ STARTHEADING
TEXT      1 "Prepared by:"
          14 "*USERNAME"
           C "The Electronics Club"
          65 "*PAGESTRING"

SUPPAGEHEADING
UNDERLINECHAR " "
TEXT      C "Actual Profit by Product"
          65 "*DATE"
TEXT      1 "Product:"
          10 "*PAGEHDR 3"

SKIP
ENDHEADING }
Profit_%
Actual

          <COLUMN (Year)
          Jan Feb Mar Apr May Jun
<ROW(Market)

{ DECIMAL 2 AFTER "%" SUPBRACKETS UNDERSCORECHAR " " }
{ INDENTGEN 1 }
<IDESCENDANTS Market
!
```

The SUPPAGEHEADING format command suppresses the default page headings from output.

The \*PAGEHDR command customizes the location of page member labels. The Sample 9 script uses page heading number 3, Product, because this is the third page dimension.

You may have also noticed that member names do not have underscores. The UNDERSCORECHAR format command blanks out underscores.

Another difference is the underlining of column headings. The `UNDERLINECHAR` format command causes the underlining to character to change to the character in quotes.

The report rows are also sorted according to their levels in the database outline. Sort commands, such as `SORTLEVEL`, do not affect individual members selected in reports. Instead, these commands work in conjunction with member selection commands.

**Note:** You can use only one sort command in a report.

Sample 9 reverses the indentation of levels from previous reports. The `INDENTGEN` command indents members to the specified number of characters.

This report script, `HEADING2.REP`, is available in your `\ESSBASE\APP\DEMO\BASIC` directory.

## Sample 10: Using Formulas

Column calculation formulas manipulate the column value of a particular row or a constant. In this report sample, each % column represents the quarterly values as a percent of Sales for the respective quarter. In addition, the Avg column represents an average value for the two quarters.

Actual Product Market					
	Qtr1	%	Qtr2	%	Avg
	=====	=====	=====	=====	=====
Sales	141,245	100.00	136,193	100.00	138,719
Cost_of_Goods_Sold	58,104	41.14	56,281	41.32	57,193
Margin	83,141	58.86	79,912	58.68	81,527
Marketing	11,211	7.94	11,302	8.30	11,257
Payroll	43,817	31.02	43,827	32.18	43,822
Miscellaneous	302	0.21	1,859	1.36	1,081
Total_Expenses	55,330	39.17	56,988	41.84	56,159
Profit	27,811	19.69	22,924	16.83	25,368
Profit_%	20	0.01	17	0.01	18
Margin_%	59	0.04	59	0.04	59

Use the following script to create Sample 10:

```
// This report performs column calculations based on values in a
// report row.
```

```
<PAGE (Scenario, Product, Market)
Actual
```

```
  <COLUMN (Year)
  Qtr1 Qtr2
```

```
{ DECIMAL 2 3 4 }
{ NAMEWIDTH 22 WIDTH 7 3 4 }
{ ORDER 0 1 3 2 4 5 }
```

```
<ROW (Accounts)
{ SAVEROW } Sales
!
```

```
{ CALCULATE COLUMN "%" = 1 % "Sales" 1 }
{ CALCULATE COLUMN "% " = 2 % "Sales" 2 }
{ CALCULATE COLUMN "Avg" = 1 + 2 / 2. }
```

```
<DESCENDANTS Accounts
!
```

**Note:** You can include comments in the report by preceding the text with `//`. The Report Extractor ignores everything that follows the double slash. You can use comments to explain report processing.

The `SAVEROW` command reserves space for a row member that the `CALCULATE COLUMN` command calculates. In this case, the calculation affects `Sales`. The `!` is required after the member name.

The `CALCULATE COLUMN` command allows column numbers, row names, or constants in formulas. You can read the first calculation this way: “% equals column 1 as a percent of Sales in column 1.”

Each calculated column label must be unique. Note how the second calculated column label has a blank space after the % sign.

To specify a constant, define a number followed by a period. You can use a constant in either a column or row calculation. The last column calculation takes the sum of columns 1 and 2 and divides by the value 2. This formula is interpreted as  $(1+2)/2$ , *not*  $1 + (2/2)$ .

As noted in Sample 7, the `ORDER` command arranges columns in the specified order. By default, calculated columns are added to the end of existing columns retrieved from the database. In this example, columns 0–2 are automatically retrieved, based on selected members. Columns 3–5 are the calculated columns. The `ORDER` command applies to both retrieved and calculated columns.

This report script, `COLCALC1.REP`, is available in your `\ESSBASE\APP\DEMO\BASIC` directory.

## Sample 11: Placing Two Page Layouts on the Same Page

This sample report has two different page layouts on the same page.

Year Profit_% Actual				
	East	West	South	Market
	=====	=====	=====	=====
Stereo	-0.52%	1.91%	0.00%	0.91%
Compact_Disc	32.60%	36.00%	0.00%	34.60%
Audio	17.86%	20.81%	0.00%	19.60%
Television	20.40%	16.57%	13.50%	17.21%
VCR	30.81%	32.43%	33.70%	32.24%
Camera	16.66%	21.66%	17.83%	19.07%
Visual	23.16%	23.56%	22.27%	23.09%
Product	21.34%	22.50%	22.27%	22.04%

Sales Actual Product					
	Qtr1	Qtr2	Qtr3	Qtr4	Year
	=====	=====	=====	=====	=====
New_York	\$18,631	\$17,681	\$19,923	\$24,403	\$80,638
Boston	\$15,812	\$15,050	\$16,716	\$19,159	\$66,737
Chicago	\$16,536	\$15,599	\$17,411	\$21,374	\$70,920
East	\$50,979	\$48,330	\$54,050	\$64,936	\$218,295
San_Francisco	\$19,761	\$19,019	\$20,722	\$24,807	\$84,309
Seattle	\$13,766	\$13,546	\$14,204	\$19,034	\$60,550
Denver	\$13,800	\$13,588	\$13,838	\$18,232	\$59,458
Los_Angeles	\$17,866	\$17,269	\$17,208	\$22,635	\$74,978
West	\$65,193	\$63,422	\$65,972	\$84,708	\$279,295
Dallas	\$ 9,226	\$ 9,175	\$ 9,481	\$12,700	\$40,582
Houston	\$ 7,690	\$ 7,363	\$ 7,646	\$10,785	\$33,484
Phoenix	\$ 8,157	\$ 7,903	\$ 8,343	\$11,843	\$36,246
South	\$25,073	\$24,441	\$25,470	\$35,328	\$110,312
Market	\$141,245	\$136,193	\$145,492	\$184,972	\$607,902

Use the following script to create Sample 11:

```
<PAGE (Year, Accounts, Scenario)

        <COLUMN (Market)
        <ICHILDREN Market

<ROW(Product)
<IDESCENDANTS Product

Actual
{ DECIMAL 2 WIDTH 10 SUPBRACKETS AFTER "%" }
Profit_%
!

<PAGE (Accounts, Scenario, Product)
Actual
Sales
Product

        <COLUMN(Year)
        <ICHILDREN Year

<ROW(Market)
{ DECIMAL 0 After " " BEFORE "$" }
<IDESCENDANTS Market
!
```

In a single report, you can select multiple dimension layouts and members. To define a multiple layout report, define reports as you normally would. Separate the commands with exclamation marks as shown above. Whenever the column, row, or page dimensions change between ! output commands, new headings are automatically generated to match the new layout.

The BEFORE format command places a character in front of data values. The AFTER format command turns off the percent signs from the first report layout.

This report script, 2LAYOUTS.REP, is available in your \ESSBASE\APP\DEMO\BASIC directory.

## Sample 12: Formatting for Data Export

This sample creates a report with a member name in each column. This format is required when you export Hyperion Essbase data to another product, such as an SQL database, with a flat file.

New York	Stereo	Sales	1000.0	950.0
New York	Stereo	Cost of Goods Sold	580.0	551.0
New York	Stereo	Margin	420.0	399.0
New York	Stereo	Marketing	80.0	80.0
New York	Stereo	Payroll	340.0	340.0
New York	Stereo	Miscellaneous	0.0	0.0
New York	Stereo	Total Expenses	420.0	420.0
New York	Stereo	Profit	0.0	-21.0
New York	Stereo	Profit %	0.0	-2.2
New York	Stereo	Margin %	42.0	42.0
New York	Compact Disc	Sales	1200.0	1150.0
New York	Compact Disc	Cost of Goods Sold	456.0	437.0
New York	Compact Disc	Margin	744.0	713.0
New York	Compact Disc	Marketing	95.0	95.0
New York	Compact Disc	Payroll	310.0	310.0
New York	Compact Disc	Miscellaneous	0.0	0.0
New York	Compact Disc	Total Expenses	405.0	405.0
New York	Compact Disc	Profit	339.0	308.0
New York	Compact Disc	Profit %	28.3	26.8
New York	Compact Disc	Margin %	62.0	62.0
New York	Audio	Sales	2200.0	2100.0
New York	Audio	Cost of Goods Sold	1036.0	988.0
New York	Audio	Margin	1164.0	1112.0
New York	Audio	Marketing	175.0	175.0
New York	Audio	Payroll	650.0	650.0
New York	Audio	Miscellaneous	0.0	0.0
New York	Audio	Total Expenses	825.0	825.0
New York	Audio	Profit	339.0	287.0
New York	Audio	Profit %	15.4	13.7
New York	Audio	Margin %	52.9	53.0
New York	Television	Sales	1800.0	1600.0

Use the following script to create Sample 12:

```
<PAGE(Scenario)

<COLUMN(Year)

<ROW (Market, Product, Accounts)
<CHILDREN East
<DESCENDANTS Product

{ DECIMAL 1
WIDTH 9
SUPBRACKETS
SUPCOMMA
MISSINGTEXT " "
UNDERSCORECHAR " "
SUPHEADING
NOINDENTGEN
SUPFEED
ROWREPEAT

Budget
    Jan Feb

<DESCENDANTS Accounts
    !
```

The ROWREPEAT command produces rows of data that have the member names repeat for each row dimension.

The SUPFEED command suppresses page feeds. A page feed automatically occurs when the report output reaches the default page length of 66 rows, unless you enter the PAGEDLENGTH command to change this setting. When a large flat file is created, you can use this command to prevent page breaks (blank rows) from being displayed in the report every time output reaches a logical page length.

This report script, FLAT2SQL.REP, is available in your \ESSBASE\APP\DEMO\BASIC directory.

## Sample 13: Creating Asymmetric Columns

Asymmetric columns make up this report. Typically, a report contains symmetric columns. That is, when multiple dimensions are displayed across the page as column groups, each level of nested columns has the same number of members nested below. Because Actual has only one nested column, Jan, and Budget has three nested columns, this report is considered asymmetric.

Some rows in the report use names other than the member names from the database. In addition to allowing aliases, as in Sample 7, you can rename a row name in the reporter.

	Product Market			
	Actual	Budget	Budget	Budget
	Jan	Jan	Feb	Mar
	=====	=====	=====	=====
Revenue	49,896	49,950	45,770	45,770
Cost of Goods	20,827	19,755	18,058	18,047
Gross Margin	29,069	30,196	27,712	27,723
Marketing	3,560	3,515	3,525	3,515
Payroll	14,599	14,402	14,416	14,416
Miscellaneous	249	0	0	0
Total Expenses	18,408	17,917	17,941	17,931
Profit	10,661	12,279	9,771	9,792

Use the following script to create Sample 13:

```
<PAGE (Product, Market)

        <COLUMN (Scenario, Year)
        Actual   Budget Budget Budget
            Jan     Jan   Feb   Mar

<ROW (Accounts)

{ RENAME "Revenue" } Sales
{ RENAME "Cost of Goods" } Cost_of_Goods_Sold
{ RENAME "Gross Margin" } Margin

{ SKIP UNDERSCORECHAR " " }
<ICHILDREN Total_Expenses

{ SKIP }
Profit
!
```

To create an asymmetric report, you must specify the member name of each column. Because the report output has two column groupings, Scenario and Year, you must specify a member from each dimension for each column. If you do not specify each column member, the resulting report format is symmetric.

The RENAME command redefines a member name when the report is output. This is useful when you do not want to use an aliases table.

This report script, ASYMM.REP, is available in your \ESSBASE\APP\DEMO\BASIC directory.

## Sample 14: Calculating Columns

This section contains two examples of CALCULATE COLUMN scripts and the reports they produce. CALCULATE COLUMN supports standard mathematical operations.

## Sample 14-A: Basic Calculated Columns

East										
Actual							Budget			Var
Jan	Feb	Mar	Qtr1			Jan	Feb	Mar	Q1	Q1
1,295	1,132	553	2,980	Tele~	Profit	1,240	950	950	3,140	(160)
25	27	14	66		Profit_%	26	22	22	70	(4)
56	62	59	177		Margin_%	60	60	60	180	(3)
1,417	1,120	898	3,435	VCR	Profit	1,466	1,161	1,161	3,788	(353)
33	30	24	87		Profit_%	35	31	31	98	(10)
61	61	62	183		Margin_%	63	63	63	189	(6)
400	272	256	928	Cam~	Profit	528	360	360	1,247	(319)
15	11	10	36		Profit_%	19	13	13	45	(10)
70	70	70	211		Margin_%	71	71	71	213	(2)
3,112	2,524	1,707	7,343	Visu~	Profit	3,234	2,471	2,471	8,175	(832)
25	24	17	66		Profit_%	27	23	23	74	(7)
61	63	63	187		Margin_%	64	64	64	191	(4)

Use the following script to create Sample 14-A:

```
<PAGE(Market)
East

      <COLUMN (Scenario, Year)
      Actual Budget
      Jan Feb Mar

{ CALCULATE COLUMN "Qtr1" = 2 : 4
  CALCULATE COLUMN "Q1" = 5 : 7
  CALCULATE COLUMN "Var~Q1" = 8 - 9

  ORDER 2,3,4,8,0,1,5,6,7,9
  WIDTH 7 WIDTH 10 0 1
}

<ROW (Product, Accounts)
<CHILDREN Visual

<CHILDREN Accounts
  !
```

This report script, COLCALC2.REP, is available in your \ESSBASE\APP\DEMO\BASIC directory.

## Sample 14-B: Asymmetric Columns

The following sample has two regular columns defined in *asymmetric* mode. For more information on asymmetric columns, see “Sample 13: Creating Asymmetric Columns” on page 37-39.

East				
Budget			Actual	Actual
Jan			Jan	% Sales
=====			=====	=====
1,200	Television	Payroll	1,236	25%
440		Marketing	365	9%
1,240		Profit	1,295	26%
4,800		Sales	5,244	100%
1,030	VCR	Payroll	1,044	25%
150		Marketing	156	4%
1,466		Profit	1,417	35%
4,200		Sales	4,311	100%
1,195	Camera	Payroll	1,167	42%
300		Marketing	288	11%
528		Profit	400	19%
2,850		Sales	2,656	100%
3,425	Visual	Payroll	3,447	29%
890		Marketing	809	8%
3,234		Profit	3,112	27%
11,850		Sales	12,211	100%

Use the following script to create Sample 14-B:

```

<PAGE(Market)
East

        <COLUMN(Scenario, Year)
        Budget  Actual
        Jan      Jan

{ ORDER 2,0,1,3,4 WIDTH 12 0 1 NOINDENTGEN AFTER "%" 4
  SKIPONDIMENSION Product LMARGIN 10 }

<ROW(Product, Accounts)

{ CALCULATE ROW "Sales" OFF }
{ CALCULATE COLUMN "Actual~% Sales" = 2 % "Sales" 2 }

<ICHILDREN Visual
{ SAVEROW } Sales
    Payroll
    Marketing
    Profit
<DUPLICATE Sales
    !

```

This report script, COLCALC3.REP, is available in your \ESSBASE\APP\DEMO\BASIC directory.

## Sample 15: Calculating Rows

The sample reports in this section demonstrate CALCULATE ROW scripts and the reports they produce.

### Sample 15-A: Basic Calculated Row

This sample report demonstrates the basic form of the CALCULATE ROW command.

	Audio Actual Sales		
	Jan	Feb	Mar
	=====	=====	=====
Boston	1,985	1,801	1,954
New_York	2,310	2,082	2,259
Chicago	2,043	1,884	1,814
Total Sales	6,338	5,767	6,027
Avg Sales	2,113	1,922	2,009

Use the following script to create Sample 15-A:

```

Audio Actual Sales
Jan Feb Mar

{ CALCULATE ROW "Total Sales" } //create new calculated row

Boston
New_York
Chicago

{ SKIP
  CALCULATE ROW "Avg Sales" = "Total Sales" /3
  PRINTROW "Total Sales"
  PRINTROW "Avg Sales" }
!
```

This report script, ROWCALC1.REP, is available in your \ESSBASE\APP\DEMO\BASIC directory.

## Sample 15-B: Calculated Rows and Missing Relationships

This sample report is a simple summary of information in a North/South grouping, which is not part of the database outline. When relationships that you need for reporting are missing in the database outline, often the best solution is to use calculated rows (or columns).

Budget Payroll			
	Jan	Feb	Mar
	====	====	====
Northern Cities			
=====			
New_York	1,940	1,930	1,930
Boston	1,610	1,610	1,610
Chicago	1,630	1,630	1,630
San_Francisco	1,815	1,815	1,815
Seattle	1,415	1,409	1,409
Southern Cities			
=====			
Denver	1,499	1,499	1,499
Los_Angeles	1,757	1,787	1,787
Dallas	1,002	1,002	1,002
Phoenix	900	900	900
Houston	834	834	834
Total Northern	8,410	8,394	8,394
Total Southern	5,992	6,022	6,022

Use the following script to create Sample 15-B:

```
// Declare Calculated Rows to Sum Southern and Northern Cities
{ CALCULATE ROW "Total Southern" OFF

// initially, set operation to OFF
  CALCULATE ROW "Total Northern" OFF }

<PAGE(Product,Scenario,Accounts)
{ RENAME "" } Product           // all products, so blank out
                                // the Product Label

Budget
Payroll
  <COLUMN(Year)
  Jan Feb Mar

<ROW(Market)                    // Northern Cities

{ SETROWOP "Total Northern" +    // Accumulate for Northern

SKIP 3
IMMHEADING// Put out heading now so text
                                // will go after it
Text 0 "Northern Cities" UCHARACTERS
}

New_York Boston Chicago San_Francisco Seattle

//Southern Cities

{ SETROWOP "Total Southern" +    } // Accumulate for Southern
{ SETROWOP "Total Northern" OFF } // Stop Accumulation for Northern

{ SKIP Text 0 "Southern Cities" UCHARACTERS }

Denver Los_Angeles Dallas Phoenix Houston

{ SKIP
PRINTROW "Total Northern"        // output calculated rows
PRINTROW "Total Southern"
}

!
```

This report script, ROWCALC2.REP, is available in your\ESSBASE\APP\DEMO\BASIC directory.

## Sample 15-C: Averaging Rows

This report sample restricts columns during calculation to average rows that contain partly numbers and percentages. The report must calculate the total regional average percentages using previously calculated rows that contain the total sales for the region. Also, the report must compute (for averaging) a count of regions. The number of regions is set as a constant in the database outline. If this number changes, the report definition must be modified. If a count of regions is not computed, a hard-to-notice error could result.

Actual Total Sales for the 3 Video Products in Qtr1: 36,914 35,126 25,119							
Budget Total Sales for the 3 Video Products in Qtr1: 37,300 34,250 26,940							
=====							
Qtr1							
Television		VCR	Camera				
		Profit	Profit_%	Profit	Profit_%	Profit	Profit_%
		=====	=====	=====	=====	=====	=====
New_York	Budget	1,020	20.40%	1,382	31.41%	540	16.68%
	Actual	847	17.66%	1,243	29.62%	352	11.79%
Boston	Budget	1,020	24.88%	1,344	35.37%	277	11.79%
	Actual	1,405	33.48%	1,002	27.49%	207	9.28%
Chicago	Budget	1,100	25.58%	1,062	31.24%	430	16.54%
	Actual	728	16.51%	1,190	30.68%	369	14.72%
San_Fran~	Budget	930	21.63%	718	21.12%	1,270	31.75%
	Actual	674	15.54%	1,197	31.12%	1,000	27.4%
Seattle	Budget	390	15.60%	973	32.98%	376	16.00%
	Actual	340	12.20%	977	31.56%	312	13.79%
Denver	Budget	690	22.26%	929	30.97%	462	18.86%
	Actual	334	11.94%	914	30.48%	361	15.92%
Los_Ange~	Budget	810	18.41%	1,101	29.76%	506	18.40%
	Actual	429	9.11%	1,127	28.81%	377	14.62%
Dallas	Budget	780	21.08%	1,341	36.24%	333	13.88%
	Actual	163	4.69%	1,055	30.28%	243	10.71%
Houston	Budget	690	24.64%	1,128	36.39%	432	18.00%
	Actual	256	10.44%	1,064	34.98%	241	10.98%
Phoenix	Budget	630	20.32%	894	31.93%	498	20.75%
	Actual	251	8.49%	940	31.07%	261	11.99%
Total Regions Averages							
Avg	Budget	806	21.61%	1,087	31.74%	512	19.02%
Avg	Actual	543	14.70%	1,071	30.49%	372	14.82%

## Use the following script to create Sample 15-C:

```

{ // Declare some of the Calculated Rows to be used
  CALCULATE ROW "Avg~Budget" OFF
  CALCULATE ROW "Avg~Actual" OFF
  CALCULATE ROW "Tot Sales~Budget" OFF
  CALCULATE ROW "Tot Sales~Actual" OFF
}

// We need the values of Market->Visual->Qtr1->Sales->Actual and
// Market ->Visual->Qtr1->Sales ->Budget to compute some
// percentages at the bottom, so get them now

Market
<CHILDREN Visual Qtr1 Sales
{ SAVEROW "Actual Sales" } Actual // stores into first 3
                                // data columns
{ SAVEROW "Budget Sales" } Budget // of these rows, which
                                // are cols 1-3
                                // change to columns 2-4 when we
                                // specify 2 row dimensions in
                                // next section

// since this is an example, not a formal report, we'll
// type out the values for Actual Sales and Budget Sales here so
// you can check the numbers:

{ SKIP 2
TEXT 0 "Actual Total Sales for the 3 Video Products in Qtr1:" 55 "*CALC"
"Actual Sales"
TEXT 0 "Budget Total Sales for the 3 Video Products in Qtr1:" 55 "*CALC"
"Budget Sales"
UCHARACTERS
SKIP 5 }
      ! // Now we can do the main report

{ AFTER "%" 3,5,7 DECI 2 3,5,7 ZEROTEXT "---" MISSING "---"
  WIDTH 10 0 1 }

```

```

<PAGE(Year)
Qtr1

        <COLUMN(Product,Accounts)
        <CHILDREN Visual
        Profit      // split these 2 accounts onto
                   // 2 lines to prevent default
        Profit_%    // to asymmetric mode
                   // because both column
                   // dimensions have the same # of
                   // members selected. Could have
                   // used <SYM instead.

<ROW(Market,Scenario)
<ONSAMELEVELAS New_York
    { SETROWOP "Avg~Actual" OFF
      SETROWOP "Avg~Budget" +

      CALCULATE ROW "Count" = "Count" + 1.  }

    Budget

    { SETROWOP "Avg~Budget" OFF
      SETROWOP "Avg~Actual" +          }

    >{ SKIP }

    Actual

{ UCOLUMNS SKIP 2 }
{
// at this point, Avg~Budget and Avg~Actual ARE NOT YET
// AVERAGES--they are the SUM of the Profit rows of each type.
// Before converting them to averages, the report computes
// Profit as a % of total sales for each type. Since we only
// have 1 value for "Budget Sales" and "Actual Sales",
// for each of the three visual products in those
// rows, the report restricts the reference to those rows to
// columns 2-4 while computing
// the percentage columns 3, 5, and 7, based on profits in
// columns 2, 4 and 6
// calculate the percentages for Budget

```

## Examples of Report Scripts

```
CALCULATE ROW "Avg~Budget" 3 = "Avg~Budget" 2 % "Budget Sales" 2
CALCULATE ROW "Avg~Budget" 5 = "Avg~Budget" 4 % "Budget Sales" 3
CALCULATE ROW "Avg~Budget" 7 = "Avg~Budget" 6 % "Budget Sales" 4

// now calculate the averages

CALCULATE ROW "Avg~Budget" 2 = "Avg~Budget" / "Count"
CALCULATE ROW "Avg~Budget" 4 = "Avg~Budget" / "Count"
CALCULATE ROW "Avg~Budget" 6 = "Avg~Budget" / "Count"

// calculate the percentages for Actual

CALCULATE ROW "Avg~Actual" 3 = "Avg~Actual" 2 % "Actual Sales" 2
CALCULATE ROW "Avg~Actual" 5 = "Avg~Actual" 4 % "Actual Sales" 3
CALCULATE ROW "Avg~Actual" 7 = "Avg~Actual" 6 % "Actual Sales" 4

// now calculate the averages

CALCULATE ROW "Avg~Actual" 2 = "Avg~Actual" / "Count"
CALCULATE ROW "Avg~Actual" 4 = "Avg~Actual" / "Count"
CALCULATE ROW "Avg~Actual" 6 = "Avg~Actual" / "Count"

TEXT C "Total Regions Averages"
PRINTROW "Avg~Budget"
PRINTROW "Avg~Actual" }
!
```

This report script, ROWAVG.REP, is available in your \ESSBASE\APP\DEMO\BASIC directory.

## Sample 16: Sorting by Top or Bottom Data Values

The following two reports demonstrate the use of TOP and BOTTOM conditional retrieval commands in a report script. For more information, see Chapter 36, “Developing Report Scripts.”

### Sample 16-A: Bottom Data Values

This sample report demonstrates the basic use of the BOTTOM command. The report is based on the Sample Basic database.

		Measures			
		Actual		Budget	
		Jan	Dec	Jan	Dec
		=====	=====	=====	=====
East	200	158	233	280	340
	300	184	277	240	210
	Diet	181	213	200	240
West	100	378	223	830	530
	300	755	971	830	950
	400	454	434	470	370
South	200	480	496	520	390
	Diet	355	404	490	430
	300	188	213	270	240
Central	300	790	824	930	810
	100	724	792	900	890
	400	691	785	660	650
Market	200	2,141	2,302	2,710	2,810
	300	1,917	2,285	2,270	2,210
	400	1,611	1,720	1,730	1,600

Use the following script to create Sample 16-A:

```
<Sym
<Column (Scenario, Year)
Actual Budget
Jan Dec
<Row (Market, Product)
<ICHILDREN Market
<ICHILDREN Product
<Bottom (3, @DataColumn(3))
!
```

The **BOTTOM** command specifies that only the three lowest data values are returned for each row grouping, based on the target data values specified in column three (Budget, Jan). Notice that no row dimension is selected here, so the report output defaults to the innermost row.

This report script, **BOTTOM.REP**, is available in your `\ESSBASE\APP\SAMPLE\BASIC` directory.

## Sample 16-B: Top Data Values

This sample report demonstrates the basic use of the TOP command. The report is based on the Sample Basic database.

		Measures			
		Actual		Budget	
		Jan	Dec	Jan	Dec
		=====	=====	=====	=====
East	Product	1,732	2,037	2,080	2,120
	100	924	1,026	960	990
	400	466	501	600	580
West	Product	2,339	2,448	2,980	2,710
	200	752	820	850	860
	Diet	663	629	850	730
South	Product	997	1,141	1,330	1,270
	100	329	432	540	640
	200	480	496	520	390
Central	Product	2,956	3,154	3,550	3,570
	Diet	1,080	1,064	1,340	1,300
	200	751	753	1,060	1,220
Market	Product	8,024	8,780	9,940	9,670
	100	2,355	2,473	3,230	3,050
	Diet	2,279	2,310	2,880	2,700

Use the following script to create Sample 16-B:

```
<Sym
<Column (Scenario, Year)
Actual Budget
Jan Dec
<Row (Market, Product)
<Ichildren Market
<Ichildren Product
<Top (10, @DataColumn(3))
!
```

The TOP command specifies that only the three highest data values are returned for each row grouping, based on the target data values specified in column three (Budget, Jan). Notice that no row dimension is selected here, so the report output defaults to the innermost row.

This report script, TOP.REP, is available in your \ESSBASE\APP\SAMPLE\BASIC directory.

## Sample 17: Restricting Rows

The following report demonstrates the use of the RESTRICT conditional retrieval command in a report script. For more information, see Chapter 36, “Developing Report Scripts.”

		Measures			
		Actual		Budget	
		Jan	Dec	Jan	Dec
		=====	=====	=====	=====
East	200	158	233	280	340
	300	184	277	240	210
	Diet	181	213	200	240
South	300	188	213	270	240
	400	#Missing	#Missing	#Missing	#Missing

Use the following script to create Sample 17:

```
<Sym
<Column (Scenario, Year)
Actual Budget
Jan Dec
<Row (Market, Product)
<Ichildren Market
<Ichildren Product
<RESTRICT (@DataCol(3) < $300.00 )
!
```

The RESTRICT command specifies that only data values that are less than \$300.00 are returned for each row grouping, based on the target data values specified in column three (Budget, Jan). Notice that no row dimension is selected here, so the report output defaults to the innermost row.

This report script, RESTRICT.REP, is available in your \ESSBASE\APP\SAMPLE\BASIC directory.

## Sample 18: Ordering Data Values

The following report demonstrates the use of the ORDERBY conditional retrieval command in a report script. For more information, see Chapter 36, “Developing Report Scripts.”

		Sales Scenario			
		Jan	Feb	Mar	Apr
		=====	=====	=====	=====
New York	100-20	#Missing	#Missing	#Missing	#Missing
	100-30	#Missing	#Missing	#Missing	#Missing
	200-20	#Missing	#Missing	#Missing	#Missing
	200-30	#Missing	#Missing	#Missing	#Missing
	300-30	#Missing	#Missing	#Missing	#Missing
	Diet	#Missing	#Missing	#Missing	#Missing
	200-10	61	61	63	66
	400-30	134	189	198	198
	300-20	180	180	182	189
	400-20	219	243	213	223
	400-10	234	232	234	245
	300-10	483	495	513	638
	200-40	490	580	523	564
	200	551	641	586	630
	400	587	664	645	666
	300	663	675	695	827
	100-10	678	645	675	712
	100	678	645	675	712
	Product	2,479	2,625	2,601	2,835

Use the following script to create Sample 18:

```
<PAGE ("Measures")
<COLUMN ("Scenario", "Year")
<ROW ("Market", "Product")
"Sales"
"Scenario"
"Jan" "Feb" "Mar" "Apr"
"New York"
```

```
"Product" "100" "100-10" "100-20" "100-30" "200" "200-10"
"200-20" "200-30" "200-40" "300" "300-10" "300-20" "300-30" "400"
"400-10" "400-20" "400-30" "Diet" "100-20" "200-20" "300-30"

<ORDERBY ("Product", @DATACOL(1) ASC, @DATACOL(2) DESC,
@DATACOL(3) ASC @DataCol (4) DESC)
!
```

The ORDERBY command is based only on data in the data columns. If the SUPPRESSMISSING command is not used in the report, #MISSING is considered to be the lowest data value. ORDERBY compares data values in the following order:

- Two values in the same column (for example, in COL1, the value associated with 200-10 is compared with the 400-30 data value, as shown in the example below).
- Data values between two data columns (for example, the data value in COL1 is compared with the data value in COL2, as shown in the example next).

If two data value are the same, the sort proceeds to the next column to determine the order.

In this example, the data value 61 in COL1 and COL2 are identical; the data in COL1 should be in ascending order, the data in COL2 should be in descending order. The two values are compared, and as they are the same, COL2 and COL3 are compared. Therefore, even though COL2 is supposed to be in descending order, the comparison for the row 400-30 was determined by the values in COL3, which is in ascending order.

	COL 1 =====	COL 2 =====	COL 3	COL 4
200-10	61	61	63	66
400-30	134	189	198	198
300-20	180	180	182	189

This report script, ORDERBY.REP, is available in your \ESSBASE\APP\SAMPLE\BASIC directory.

## Sample 19: Narrowing Member Selection Criteria

The following report demonstrates the use of the LINK command to narrow the members returned in a selection in a report script. For more information, see Chapter 36, “Developing Report Scripts.”

Market Measures Scenario		
	Qtr1	Qtr2
	=====	=====
100-10	5,096	5,892
100-20	1,359	1,534
100-30	593	446
200-10	1,697	1,734
200-20	2,963	3,079
200-30	1,153	1,231
200-40	908	986
300-10	2,544	3,231
300-20	690	815
300-30	2,695	2,723
400-10	2,838	2,998
400-20	2,283	2,522
400-30	(116)	(84)
100-20	1,359	1,534
200-20	2,963	3,079
300-30	2,695	2,723
Product	24,703	27,107

Use the following script to create Sample 19:

```
<PAGE (Market)
<COLUMN (Year)
Qtr1 Qtr2
<ROW (Product)
<LINK (<UDA (product, naturally-flavored) OR <LEV (product, 0))
!
```

The LINK command uses the AND, OR, and NOT Boolean operators to refine your search. In the preceding example, the product with the “naturally-flavored” user-defined attribute, as well as all Level 0 products, are returned in the search.

Be careful how you group operators in the LINK expression. Hyperion Essbase evaluates operators from left to right. Use parentheses to group the expressions. For example, A OR B AND C is the same as ((A OR B) AND C). In the first expression, Hyperion Essbase evaluates the expression from left to right, evaluating A OR B before evaluating AND C. In the second expression, Hyperion Essbase evaluates the subexpression in parentheses (A OR B) before the whole expression, producing the same result. However, if you use (A OR (B AND C)), Hyperion Essbase evaluates the subexpression in parentheses (B AND C) before the whole expression, producing a different result.

This report script, LINK.REP, is available in your \ESSBASE\APP\SAMPLE\BASIC directory.

## Sample 20: Using Attributes in Member Selection

This sample report uses members of attribute dimensions to view data on base dimensions that are associated with those attribute dimensions.

	Profit Actual Caffeinated_True Qtr1 East				
	Ounces_32	Ounces_20	Ounces_16	Ounces_12	Ounces
	=====	=====	=====	=====	=====
Bottle	#Missing	488	240	(586)	142
Can	#Missing	#Missing	#Missing	2,776	2,776
Pkg Type	#Missing	488	240	2,190	2,918

Use the following script to create Sample 20:

```
{WIDTH 12}
<PAGE (Measures, Scenario, Caffeinated, Year, Market)
Profit
Actual
Caffeinated_True
Qtr1
East
<COLUMN (Ounces)
<CHILDREN Ounces
<ROW ("Pkg Type")
<CHILDREN "Pkg Type"
!
```

The report output reflects data on Quarter 1 profits for caffeinated products by all their available sizes and package types. The data values indicate #MISSING when there is no data for a specific size in a specific package type. Because attributes are defined only on sparse dimensions, there are several #MISSING values in the sample report. You can represent missing values by suppressing the row or substituting a replacement text string, such as N/A. See “Sample 2: Handling Missing Values” on page 37-4 for an example of substituting page breaks and labels for missing values.

This report script, ATTR.REP, is available in your \ESSBASE\APP\SAMPLE\BASIC directory.

## Sample 21: Using the WITHATTR Command in Member Selection

This sample report uses the WITHATTR command to view information based on the attributes of the members of a base dimension.

	Profit Actual Qtr1 East		
	Bottle	Can	Pkg Type
	=====	=====	=====
100-30	74	#Missing	74
200-30	#Missing	#Missing	#Missing
200-40	908	#Missing	908
400-10	645	#Missing	645
400-20	290	#Missing	290
400-30	545	#Missing	545

Use the following script to create Sample 21:

```
{WIDTH 12}
<PAGE (Measures, Scenario, Year, Market)
Profit
Actual
Qtr1
East
<COLUMN ("Pkg Type")
<CHILDREN "Pkg Type"
<ROW (Product)
<WITHATTR(Caffeinated,"<>",True)
<IDESCENDANTS Product
!
```

The report output reflects data on Quarter 1 profits for caffeinated products by their package types. The data values indicate #MISSING when there is no data for a specific package type. Because attributes are defined only on sparse dimensions, there are several #MISSING values in the sample report.

This report script, WITHATTR.REP, is available in your \ESSBASE\APP\SAMPLE\BASIC directory.

Chapter  
**38**

# Optimizing Your Reports

---

The time required to generate a report varies depending upon the size of the database you are reporting from, the number of report queries included in the script, the size of the report buffer, and other factors.

This chapter describes ways to optimize the time required to generate your reports. It discusses the following topics:

- “Setting Configurable Variables” on page 38-2
- “Generating Symmetric Reports” on page 38-6
- “Organizing Members to Optimize Data Extraction” on page 38-7
- “Working with Database Outlines” on page 38-8

If you are migrating from a previous version of Hyperion Essbase, see the *Hyperion Essbase Start Here* booklet for important information about system changes and enhancements.

## Setting Configurable Variables

Configurable variables specify the size of the internal server buffers used for storing and sorting data extracted by a report. The following report variables are used in the conditional retrieval and data sorting commands:

- Retrieval Buffer Size setting
- Retrieval Sort Buffer Size setting
- NUMERICPRECISION setting

For more information about these settings, see the online *Technical Reference* in the DOCS directory. For information about the Report Extractor's process of retrieving data, see Chapter 35, "Quick Start to Report Scripts."

## Setting the Retrieval Buffer Size

The Retrieval Buffer Size setting specifies the size, in kilobytes, of the server buffer that holds extracted row data cells before they are evaluated by the RESTRICT, TOP, or BOTTOM commands. If this buffer is full, the rows are processed, and the buffer is reused. Both the Spreadsheet Retrieval Wizard and the Report Writer use this buffer to process retrievals.

You can adjust the buffer size on a per database basis. The default buffer size is set to 10 kilobytes. If you are increasing the size of the buffer, it is recommended that you do not exceed 100 kilobytes, although the size limit is set at 100,000 kilobytes. Larger buffer sizes can create problems when concurrent users generate reports at the same time.

- To set the retrieval buffer size:
  1. From the Application Manager menu, choose Database > Settings.
  2. Select the General page.
  3. Type the number, in kilobytes, of the required buffer size in the Retrieval Buffer Size box.

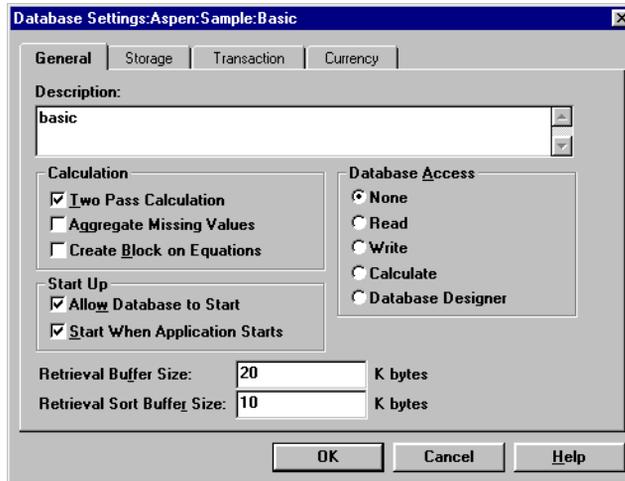


Figure 38-1: Setting the Retrieval Buffer Size

**Note:** To determine the best buffer size for your report processing needs, test a report script with different settings.

In Hyperion Essbase Version 4 this variable was called REPTKBYTE and was set in the `ESSBASE.CFG` file. You can no longer set the variable in the `ESSBASE.CFG` file.



You can use the `SETDBSTATEITEM` command in `ESSCMD` to set the retrieval buffer size. See the online *Technical Reference* in the `DOCS` directory for information about this command. See Chapter 44, “Performing Interactive and Batch Operations Using `ESSCMD`” for information about `ESSCMD`.

## 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 an Hyperion Essbase Spreadsheet Add-in or Application Manager Report Writer retrieval. If the sorting buffer is full, Hyperion Essbase posts an error message.

You can adjust the buffer size on a per-database basis. The default buffer size is set to 10 kilobytes.

- To set the retrieval sort buffer size:
  1. From the Application Manager menu, choose Database > Settings.
  2. Select the General page.
  3. Type the number, in kilobytes, of the required sort buffer size in the Retrieval Sort Buffer Size box.

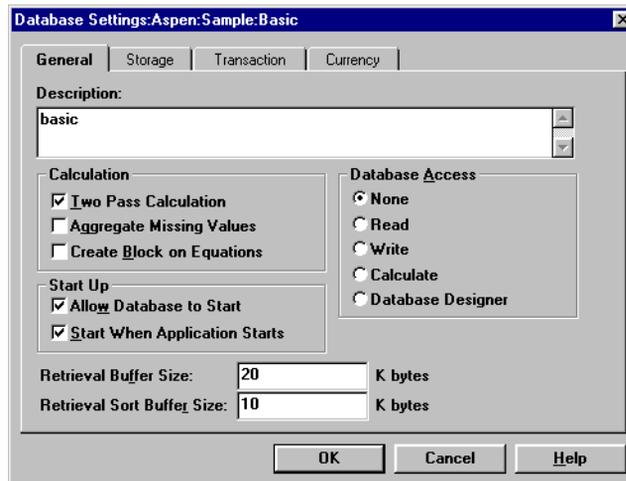


Figure 38-2: Setting the Retrieval Sort Buffer Size

**Note:** To determine the best buffer size for your report processing needs, test a report script with different settings.

In Hyperion Essbase Version 4, this variable was called REPTKBYTESORTBUF and was set in the `ESSBASE.CFG` file.



You can use the SETDBSTATEITEM command in ESSCMD to set the retrieval sort buffer size. See the online *Technical Reference* in the DOCS directory for information about this command. See Chapter 44, “Performing Interactive and Batch Operations Using ESSCMD” for information about ESSCMD.

## Setting the NumericPrecision Variable

The NUMERICPRECISION setting, used by the RESTRICT command, defines the number of precision digits the internal numerical comparison considers in the Report Extractor.

► To set the NUMERICPRECISION variable:

1. Open the `ESSBASE.CFG` server file.

**Note:** The `ESSBASE.CFG` file is created manually. Consult your Database Administrator if the file is not available.

2. Find the following line in the file:

```
NUMERICPRECISION X
```

where *X* is a numeric value representing the number of precision digits that are used by the Report Writer for numerical comparison.

**Note:** If the line does not exist, type in the variable information using the example above. See the online *Technical Reference* in the DOCS directory for a syntax description and details about the NUMERICPRECISION setting, as well as more information about adding information to the `ESSBASE.CFG` file.

## Generating Symmetric Reports

If report processing time is of primary importance, consider making all reports symmetric. Symmetric reports provide better processing performance than asymmetric reports, as the Report Extractor composes the member list based on all possible member combinations, which it can do in one pass. With asymmetric reports, the Extractor must retrieve and process each block of possible member combinations separately.

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

**Block**

Figure 38-3: Symmetric Report Member Combinations

Sales South				
	Actual		Budget	
	Jan		Feb	
	=====		=====	
100-10	757		950	
100-20	450		590	
100-30	#Missing		#Missing	
100	1,207		1,540	

**Block 1      Block 2**

Figure 38-4: Asymmetric Report Member Combinations

For more information about how the Report Extractor retrieves data, see Chapter 35, “Quick Start to Report Scripts.”

## Organizing Members to Optimize Data Extraction

Report Extractor extracts data in a certain order. If you do not require a formatted report, you can reduce the time required to generate the report by:

- Structuring the report script in the same order as Report Extractor extracts data
- Grouping dense dimensions in columns
- Grouping sparse dimensions in rows

This is particularly useful when creating 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 then proceeding from the innermost row member (right) to the outermost row member (left). In Figure 38-5, the report is read in numerical sequence.

Sales South		
	Actual	Budget
	Jan	Jan
	-----	-----
100-10	757	950
100-20	450	590
100-30	#Missing	#Missing
100	1,207	1,540

Figure 38-5: How the Report Extractor Examines Data

In Figure 38-5, the column members come from dense dimensions, and the row members come from sparse dimensions. To optimize the time needed to extract data, group dense dimensions first, then group sparse dimensions in the same sequence as they are displayed in your outline.

By nesting dense dimensions in the report columns, Report Extractor examines each block only once, thus improving the performance time. Sparse dimensions follow the dense dimensions in the outline.

**Note:** Attributes are sparse dimensions and are dynamically calculated. Hence, Hyperion Essbase cannot use the sparse data extraction method when a report contains attribute dimensions. For more information on the SPARSE command, see the online *Technical Reference* in the DOCS directory.

## Working with Database Outlines

Report scripts extract data based on members identified in the database outline. This section discusses time implications when generating reports.

### Accessing Outlines that Contain Dynamic or Transparent Members

If you generate a report that accesses a database outline containing 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 containing Dynamic Calc or Dynamic Time Series members, Hyperion Essbase calculates the outline member every time a report is generated, which increases the reporting time.

For more information about dynamic members, see Chapter 29, “Dynamically Calculating Data Values.”

If you run a report that contains transparent members, the report takes longer to generate, as it must access more than one server to retrieve the required data.

### Ensuring that Members in the Report Script Synchronize with the Database Outline

When you run a report, it is important to ensure that the member names referenced in the report script match the members in the database outline. An error displays every time the Report Extractor cannot find a matching member name, and you must correct the name in the report script before the report continues processing.

# Copying Data Subsets and Exporting Data to Other Programs

---

You can move data between Hyperion Essbase OLAP Server databases or to another program by extracting an output file of the data you want to move. For example, you can copy a subset of an Hyperion Essbase database from the Hyperion Essbase server to Personal Essbase.

In order to meet the import format specifications of most other programs, use the Report Writer to create a text file.

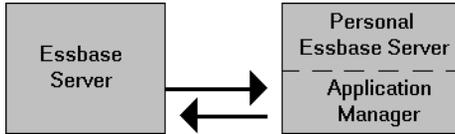
This chapter contains information about the following topics:

- “Copying a Database Subset to Personal Essbase” on page 39-1
- “Using Report Scripts for Data Exporting” on page 39-13

## Copying a Database Subset to Personal Essbase

This version of Hyperion Essbase includes the ability to install both the Hyperion Essbase server and client on a Windows NT workstation or a Windows 95 or Windows 98 machine. This is a one-port license and has its own license number. For information about installing and configuring Personal Essbase on a machine, see the *Hyperion Essbase Installation Guide*.

Once you have installed Personal Essbase, you can copy the outline file (*dbname.OTL*) and a data subset from the Hyperion Essbase server and load them into Personal Essbase. The Personal Essbase server does not communicate with the Hyperion Essbase server.



*Figure 39-1: Hyperion Essbase Server and Personal Essbase Interaction*

**Note:** Do not create more than one application and two databases on the Personal Essbase server.

## Summary of Steps

To copy a database subset to Personal Essbase, complete these steps. The following sections provide detailed information on completing each step.

1. On your Personal Essbase server, create a new application and database to contain the database subset.
2. Copy the outline file (for example, *source\_dbname.OTL*) from your source database to the new database on Personal Essbase.

You may need to rename the outline file to match the name of your Personal Essbase database (for example, *target\_dbname.OTL*), overwriting the existing target database outline file.

3. Create an output file (for example, an ASCII text file) containing the required data subset.
4. Load the output file into the new database that you have created.

See Chapter 20, “Introducing Data Loading” for more information about loading data.

If required, you can repeat steps 3 and 4 to create an output file from the database on your Personal Essbase server and load the data back into the main Hyperion Essbase database on a different machine.

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 a Personal Essbase server.

## Creating a New Application and Database

Create a new application and database on your Personal Essbase server. You will copy the required subset of data into this new database. You can give this application and database any name. In this example, you copy a subset of data for the West market, and name the application `WEST.APP`, and the database `WESTMKTS.DB`.

- To create the application and database:
  1. In Application Manager, choose `File > New > Application` to open the Create New Application dialog box. Type the new application name in the Application name text box. For example, type **West**.
  2. Click OK.
  3. In Application Manager, choose `File > New > Database` to open the Create New Database dialog box. Type the new database name in the Database name text box. For example, type **Westmkts**.
  4. Click OK.

Hyperion Essbase creates the new application and database.

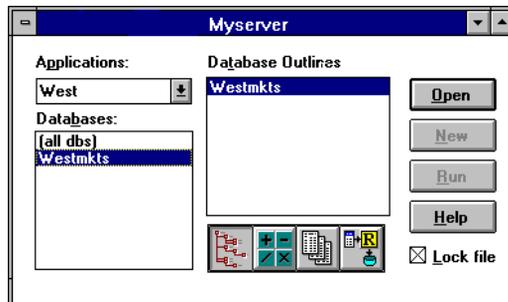


Figure 39-2: New West Application and Westmkts Database

Ensure that the new, empty database is not running. In Application Manager, select the new database and choose **Application > Start/Stop** from the main menu. A message box tells you if the database is running. If necessary, click **Yes** in the Stop database message box to stop the database.



Figure 39-3: Stop Database Dialog Box

## Copying the Outline File from Your Source Database

Copy the outline file (.OTL) of your 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 to `WESMKTS.OTL` on Personal Essbase.

How you copy the outline file depends on whether you can connect to the source Hyperion Essbase database from your Personal Essbase machine.

- If you *can* connect, use **File > Save As** in Application Manager to copy the outline. See “Copying the Outline File Using Application Manager” on page 39-5.
- If you *cannot* connect (for example, if your Personal Essbase machine is a laptop machine that has no network connection), use your operating system to copy the outline file. For more information, see “Copying the Outline File Using the Operating System” on page 39-6.



You can use the `COPYDB` command in `ESSCMD` to perform this task. See the online *Technical Reference* in the `DOCS` directory for information about this command. See Chapter 44, “Performing Interactive and Batch Operations Using `ESSCMD`” for information about `ESSCMD`.

## Copying the Outline File Using Application Manager

Complete these steps if you *can* connect to the source Hyperion Essbase database from your Personal Essbase machine.

1. Connect to the source server.
2. In Application Manager, open the source outline. For example, open Sample Basic.

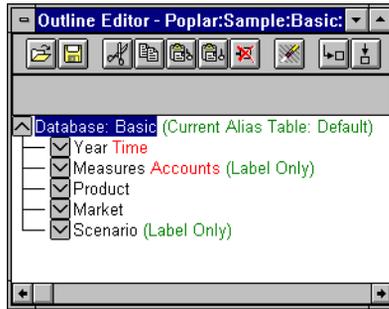


Figure 39-4: Sample Basic Outline

3. Choose File > Save As from the Outline Editor menu. The Save Server Object dialog box displays.

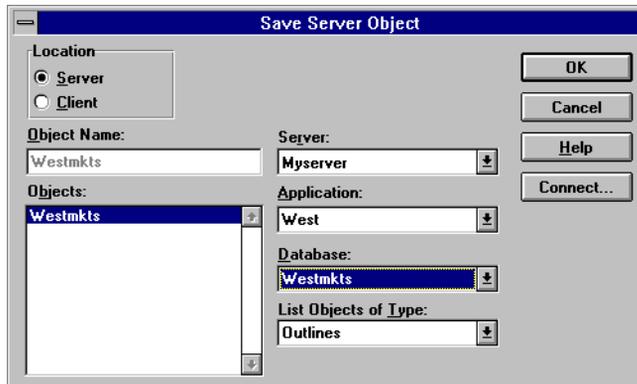


Figure 39-5: Save Server Object Dialog Box

4. In the Server, Application, and Database list boxes, select the new application and database that you have just created on the Personal Essbase server.
5. Click OK in the Save Server Object dialog box to copy the outline file. At the prompt, click Yes to replace the existing WESTMKTS.OTL file. The existing outline file is an empty file, which Hyperion Essbase created automatically when you created the new application and database.
  - Hyperion Essbase might prompt you to unlock and overwrite the file. If necessary, click Yes at the prompts.
  - Hyperion Essbase might warn you that the outline file did not originate from the latest version on the server. If necessary, click Yes to continue.

## Copying the Outline File Using the Operating System

Complete these steps if you *cannot* connect to the source Hyperion Essbase database from your Personal Essbase machine.

1. Use the operating system to copy the source outline file. For example, copy BASIC.OTL to WESTMKTS.OTL. Give the copied outline exactly the same name as the new database.
2. Save the outline file in the `\ARBORPATH\APP\appname\dbname` directory on your Personal Essbase machine, where *ARBORPATH* is the directory in which your installed Hyperion 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 `C:\ESSBASE\APP\WEST\WESTMKTS\WESTMKTS.OTL` on your Personal Essbase machine. 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 Hyperion Essbase created automatically when you created the new application and database.

3. Before using the new outline file, you need to stop and restart the new database. Select the new database in Application Manager. For example, select `Westmktts`.
4. Choose `Application > Start/Stop` from the main menu to stop the new database, and choose `Application > Start/Stop` again to restart the database.

You now have a copy of the database outline on your Personal Essbase server.

## Creating an Output File Containing the Required Data Subset

Create an output file that contains the required data subset. The output file can be an ASCII text file, or a spreadsheet file. You can use the Report Writer to create an ASCII text file of the data subset. For example, use `<IDESCENDANTS` to select a data subset.

**Note:** You can also use the Spreadsheet Client Retrieval Wizard to create a spreadsheet file of the data subset. For more information on using the Retrieval Wizard, see the *Hyperion Essbase Spreadsheet Add-in User's Guides*.

- To create an ASCII text file that contains the required data subset:
  1. In Application Manager, select the source database. For example, select `Westmktts`.

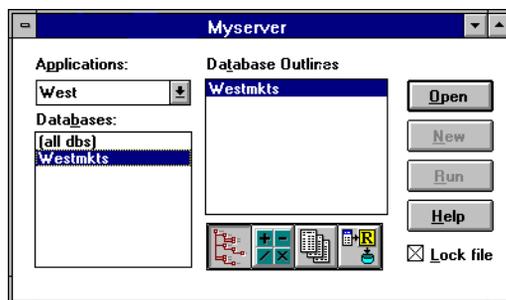


Figure 39-6: `Westmktts` Database Selected in Application Manager

- If you *can* connect to your main Hyperion Essbase database from your Personal Essbase machine, you can select the source database from your Personal Essbase machine.
  - If you *cannot* connect, use a different machine from your Personal Essbase machine to select the source database.
2. Click the Report icon, , and then click New to open the Report Editor.
  3. Write a report script that selects the required data subset. For information on writing report scripts, see Chapter 35, “Quick Start to Report Scripts” and the online *Technical Reference* in the DOCS directory.

For example, the following report script selects the Actual, Measures data for the West market from Sample Basic:

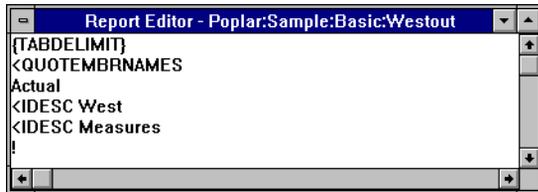


Figure 39-7: Report Editor With Sample Basic Report Script

- Use TABDELIMIT to place tab stops between data, instead of spaces. This ensures that no member names or data values are truncated.
- Use QUOTEMBRNAMES to place quotation marks (") around member names that contain blank spaces. Hyperion Essbase then recognizes the member names when it loads the data.

4. In the Report Editor, choose Report > Output Options to open the Report Output Options dialog box. Check the File and Window check boxes.

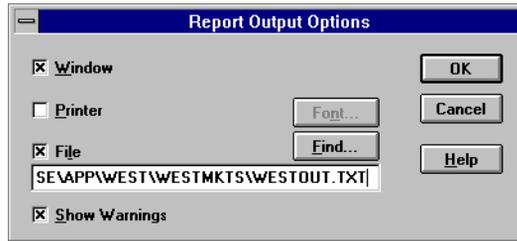


Figure 39-8: Report Output Options Dialog Box

5. Give the report output file any name with a .TXT extension. For example, WESTOUT.TXT.

To load the data, the output file needs to be in the `\ARBORPATH\APP\appname\dbname` directory on your Personal Essbase server, where `ARBORPATH` is the directory in which you installed Hyperion Essbase, and `appname` and `dbname` are the new application and database directories that you have created.

If you are using your Personal Essbase machine, you can save the output file directly into the `\ARBORPATH\APP\appname\dbname` directory. For example, type `C:\ESSBASE\APP\WEST\WESTMKTS\WESTOUT.TXT` in the File text box.

If you are *not* using your Personal Essbase machine, save the output file anywhere on the current machine. By default, Hyperion Essbase saves the file on your Hyperion Essbase client machine, and not on the Hyperion Essbase server. When you run the report, use your operating system to copy the file to the `\ARBORPATH\APP\appname\dbname` directory on your Personal Essbase server. For example, use a disk to copy the file.

6. Click OK in the Report Output Options dialog box.

7. Save your report script and choose Report > Run from the Report Editor menu to run the report.

Hyperion Essbase displays the report in a window and sends it to the file you specified in the Report Output Options dialog box.

```

Report Viewer - Poplar:Sample:Basic:Westout
Year Product Actual
California Oregon Washington Utah Nevada West
Sales 47,442 19,992 19,036 17,305 29,156 132,931
COGS 22,741 8,647 7,907 7,883 15,727 62,905
Margin 24,701 11,345 11,129 9,422 13,429 70,026
Marketing 8,031 3,013 3,622 2,835 6,028 23,529
Payroll 3,639 3,201 2,802 3,363 3,297 16,302
Misc 67 69 64 69 65 334
Total Expenses 11,737 6,283 6,488 6,267 9,390 40,165
Profit 12,964 5,062 4,641 3,155 4,039 29,861
Opening Inventory 11,265 5,971 5,839 5,756 9,920 38,751
Additions 55,948 20,925 19,771 19,405 37,753 153,802
Ending Inventory 19,771 6,904 6,574 7,856 18,517 59,622
Inventory 11,265 5,971 5,839 5,756 9,920 38,751
Margin % 52 57 58 54 46 53
Profit % 27 25 24 18 14 22
Ratios 52 57 58 54 46 53
Measures 12,964 5,062 4,641 3,155 4,039 29,861
    
```

Figure 39-9: Sample Basic Report Script Output

If you are *not* using your Personal Essbase machine, remember to download and copy the file from the Hyperion Essbase client directory to the `\ARBORPATH\APP\appname\dbname` on your Personal Essbase server. For example, copy the output file to `C:\ESSBASE\APP\WEST\WESTMKTS\WESTOUT.TXT`.

You are now ready to load the text file into your new database.

## Loading the Output File Into the New Database

Load the output file into the new database on your Personal Essbase machine.

1. In Application Manager on your Personal Essbase server, select the new database. For example, select `Westmkt.s`.

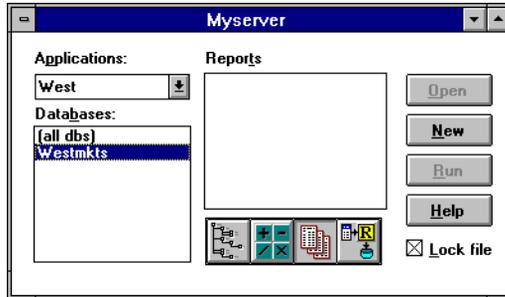


Figure 39-10: *Westmkt.s* Database Selected in Application Manager

2. Choose Database > Load Data from the Application Manager main menu.  
The Data Load dialog is box displayed.
3. Click Find to open the Open Server Data File Objects dialog box.
  - In the Objects list box, select the ASCII text file you have just created. For example, select `WESTOUT`.
  - Click OK.

**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 “Creating an Output File Containing the Required Data Subset” on page 39-7.

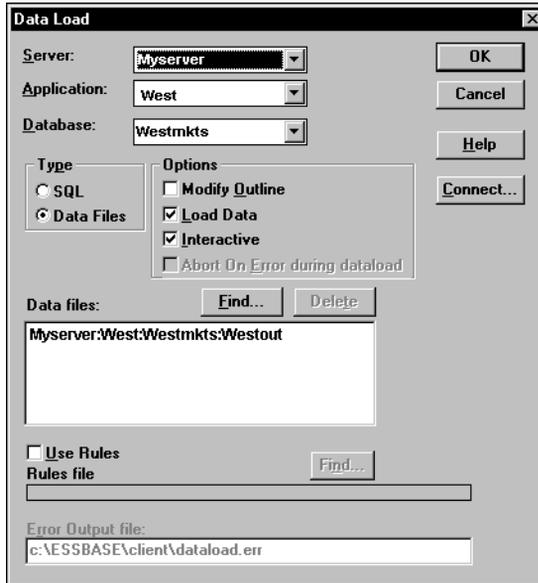


Figure 39-11: Data Load Dialog Box with WESTOUT Selected

Click OK in the Data Load dialog box to load the text file into the new database. For detailed information on loading data and any errors that may occur, see Chapter 20, “Introducing Data Loading.”

You can now view the data on your Personal Essbase machine. You might need to recalculate the database subset. Because you are viewing a subset of the database, a percentage of the data values will be #MISSING.

If required, you can copy report scripts and other object files to your Personal Essbase machine to use with the database subset you have created.

## Using Report Scripts for Data Exporting

You can use report scripts to export Hyperion Essbase data to other programs in text format. Report Writer enables you to create text files that meet the import format specifications of most other programs.

Before you can import data into some programs, you must separate, or delimit, the data with specific characters.

If you plan to import Hyperion Essbase data into a program that requires special delimiters, use the MASK command. For the syntax and usage of Report Writer commands, see the online *Technical Reference* in the DOCS directory.

**Note:** You cannot export data generated by Dynamic Calc members. Because attributes are Dynamic Calc members, you cannot export data generated by attributes.

When you export data to a program that uses a two-dimensional, fixed-field format, you do not need to 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}
<CHILDREN Year
Sales
<CHILDREN "400"
East
Budget
!
```

This example produces the following 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

If you want 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 you want to print.
- To list only level 0 data for a given member (including a dimension), use the **DIMBOTTOM** command as a parameter with the dimension name that contains the data 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
!
```

This example produces the following 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

See Chapter 37, “Examples of Report Scripts,” for another example of formatting for data export.

## Importing Data Into Other Databases

Before you import data into some programs, you must delimit the data with specific characters. If you plan to import Hyperion Essbase data into a program that requires special delimiters, use the MASK command. For the syntax and usage of Report Writer commands, see the online *Technical Reference* in the DOCS directory.

Part  
**VII**

# Managing Multidimensional Hyperion Essbase Data Storage

Part VII describes how to manage data storage in Hyperion Essbase OLAP Server databases, including how to use the Hyperion Essbase Storage Manager and ensure data integrity. Part VII contains the following chapters:

- Chapter 40, “Introducing the Hyperion Essbase Kernel,” introduces you to the Hyperion Essbase Storage Manager and describes how to use it to retrieve data, compress data, allocate storage, handle fragmentation, and restructure databases.
- Chapter 41, “Specifying Hyperion Essbase Kernel Settings,” describes how to set Storage Manager settings to specify the index cache size, the data cache size, the index page size, the isolation level, the disk volumes, and data compression using the Hyperion Essbase Application Manager and ESSCMD.
- Chapter 42, “Ensuring Data Integrity,” introduces you to transactions, isolation levels, data redundancy, and how Hyperion Essbase locks data, commits data, rolls back transactions in the event of a server interruption, and checks structural integrity.



Chapter  
**40**

# Introducing the Hyperion Essbase Kernel

This chapter describes how Hyperion Essbase accesses and stores data.

This chapter contains the following sections:

- “About the Hyperion Essbase Kernel” on page 40-1
- “Data Compression” on page 40-8
- “Storage Allocation” on page 40-11
- “Fragmentation” on page 40-14
- “Database Restructuring” on page 40-14

## About the Hyperion Essbase Kernel

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

The Hyperion Essbase kernel has the following functions:

- Handles disk storage and caching of Hyperion Essbase files
- Handles data retrieval
- Handles data updates
- Controls input-output functions related to Hyperion Essbase
- Consolidates free space for re-use
- Manages concurrent operations

- Recovers databases after a server crash
- Issues locks
- Manages transactions

The Hyperion Essbase kernel uses direct I/O (input/output) and no-wait I/O on most server platforms and file systems (for a list of the supported platforms, see the *Hyperion Essbase Installation Guide*). This means that Hyperion Essbase caches index and data files (.IND and .PAG files) directly and is able to perform asynchronous, overlapped I/Os. The following benefits are provided:

- Faster response time. A user waits less time for Hyperion Essbase to return data.
- Scalability and predictability. You determine the optimal cache sizes for Hyperion Essbase databases.

The Hyperion Essbase kernel contains components that control all aspects of retrieving and storing data. The following table summarizes the functions of these components:

*Table 40-1: Hyperion Essbase Kernel Components*

<b>Component</b>	<b>Main Function</b>
Index Manager	Find and track the location of requested data
Data Block Manager	Retrieves the data pointed to by the index and stores the data
LRO Manager	Handles retrieval and storage of linked reporting objects (LROs)
Lock Manager	Handles the locking of data blocks to regulate concurrent data access
Transaction Manager	Tracks transactions and handles internal commit and abort operations

For information on how the client communicates with the server, see Chapter 45, “Running Hyperion Essbase, Applications, and Databases.”

## Index Manager

The Index Manager manages the database index and provides a fast way of looking up Hyperion Essbase data blocks. The Index Manager manages the index cache and determines which portions of the database index are to be cached in the index cache.

The Index Manager controls five components. The following table summarizes the functions of these components:

Component	Main Function
Index	Contains all index files.
Index File	Stores retrieval information in index pages. Hyperion Essbase names index files incrementally on each disk volume, using the naming convention <code>ESSxxxxx.IND</code> , where <code>xxxxx</code> is a number. The first index file on each disk volume is named <code>ESS00001.IND</code> .
Index page	Holds index entries that point to data blocks.
Index entry	Points to a data block. An index entry exists for every intersection of sparse dimensions.
Index cache	Holds multiple index pages in memory.

The term *index* refers to all index files for a single database. The index can span multiple volumes, and more than one index file can reside on a single volume. Use the disk volumes setting to specify disk spanning parameters. See Chapter 41, “Specifying Hyperion Essbase Kernel Settings,” for information on setting index cache size and index page size and for information on allocating space with the disk volumes setting.

## Allocation Manager

The Allocation Manager is part of the Index Manager. The Allocation Manager manages:

- 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

The Allocation Manager bases space allocations on the disk volumes setting. You can specify disk volumes in Hyperion Essbase Application Manager or with ESSCMD. See Chapter 41, “Specifying Hyperion Essbase Kernel Settings,” for information on the disk volumes setting.

The Allocation Manager makes decisions about when and where to create a new file. When additional data needs to be stored, the Allocation Manager proceeds as follows:

1. It attempts to use free space in an existing file.
2. If enough free space is not available, it attempts to expand an existing file.
3. If enough free space is not available in existing files, it creates a new file on the current volume.
4. If it cannot expand an existing file or create a new 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.

## 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 the functions of these components:

Component	Main Function
Data file	Stores retrieval information in data blocks. Hyperion Essbase names data files incrementally: <code>ESSxxxxx.PAG</code> , where <code>xxxxx</code> is a number starting with 00001.
Data block	A multidimensional array that represents cells of the dense dimensions for a given intersection of sparse dimensions.
Data cache	Holds data blocks in memory.
Data file cache	Holds data files in memory.

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; more than one database can reside on the same volume. See Chapter 41, “Specifying Hyperion Essbase Kernel Settings,” for information on specifying data file cache size and data cache size and on allocating disk space.

## LRO Manager

Linked reporting objects (LROs) enable you to associate objects, such as flat files, with data cells.

The Hyperion Essbase kernel stores information about linked reporting objects in an LRO catalog. Each catalog resides in its own Hyperion Essbase index page and coexists in an index file (`ESSxxxxx.IND`) with other, non-LRO Hyperion Essbase index pages. Users can create and store LRO files, with an `.LRO` extension, in the database directory (`\ESSBASE\applicationname\database\name`, for example, `ESSBASE\SAMPLE\BASIC`).

For more information about linked reporting objects, see Chapter 12, “Linking Objects to Hyperion Essbase Data,” and the *Hyperion Essbase Spreadsheet Add-in User’s Guide*.

## Lock Manager

The Lock Manager issues locks on data blocks and 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 42, “Ensuring Data Integrity.”

## Transaction Manager

The Transaction Manager controls transactions and commit operations and manages database recovery.

Hyperion Essbase commits data automatically. Commits are triggered by transactions that modify data: data loading, calculating, restructuring, and spreadsheet lock&send operations.

How Hyperion Essbase commits data depends upon whether the transaction isolation level is set to committed or uncommitted access (the default). See Chapter 41, “Specifying Hyperion Essbase Kernel Settings,” for information on specifying isolation level.

The Transaction Manager maintains a transaction control table, *dbname.TCT*, to track transactions.

See Chapter 42, “Ensuring Data Integrity,” for more information on commit operations and recovery.

## Hyperion Essbase Kernel Startup

This section provides a summarized, high-level list of the steps that lead to Hyperion Essbase kernel start-up.

1. A user starts a database.
2. Hyperion Essbase loads the database.
3. The Hyperion Essbase Agent passes database settings to the server.
4. The Hyperion Essbase kernel begins its initialization process.
5. The Hyperion Essbase 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 Hyperion Essbase kernel shuts itself down.

## Fatal Error Handling

When the Hyperion Essbase kernel encounters a fatal error, it shuts down and restarts, attempting to reinitialize itself and proceed with database recovery. When recovery begins, Hyperion Essbase displays an error message similar to the following:

“Reinitializing the Hyperion Essbase kernel due to a fatal error.”

This message is followed by other informational messages related to database recovery, such as the following:

“Invalid transaction handle passed to the Hyperion Essbase kernel.”

When you see these messages, you know that the Hyperion Essbase kernel shut itself down and is attempting to start up. Check the server log and determine whether Hyperion Essbase issued a fatal error message just before generating the reinitialization messages. See Chapter 46, “Using Diagnostics to Monitor Performance,” for information about the server log. See “Rolling Back Transactions” on page 42-11 for more information about recovering transactions.

If the Hyperion Essbase kernel did encounter a fatal error, in most cases you need to reinitiate 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 server log to see how far Hyperion Essbase processed the operation. When in doubt, reinitiate the operation.

If the Hyperion Essbase kernel did not encounter a fatal error, contact your software provider's technical support to determine what caused the Hyperion Essbase kernel to shut down and restart.

The Hyperion Essbase kernel considers the following errors fatal:

- One or more control fields have unexpected or inconsistent values.
- The Hyperion Essbase kernel detects data corruption.
- The Hyperion Essbase kernel cannot perform an operation that is necessary to ensure data integrity (for example, disk space is insufficient).
- The Hyperion Essbase kernel encounters a condition that can lead to data corruption.

## Data Compression

A data compression setting determines whether Hyperion Essbase compresses data blocks when it writes them out to disk. Hyperion Essbase fully expands the compressed data blocks, including empty cells, when the blocks are swapped into the data cache. Data compression is enabled by default. Generally, data compression optimizes storage use.

There are two types of data compression: bitmap and run-length encoding (RLE).

### Using Bitmap Data Compression

With bitmap compression, Hyperion 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 non-missing. When a data block is not compressed, Hyperion Essbase uses eight bytes to store every non-missing cell.

When using bitmap compression, Hyperion Essbase stores only non-missing values and does not compress repetitive values or zeros (contrast with RLE compression, described in “Using RLE Data Compression” on page 40-9). When Hyperion Essbase pages 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 40-1 represents a portion of a data block, as an example. In this example, Hyperion Essbase uses 64 bytes to

store the data in the fully expanded block, but uses one byte (eight bits) to store the bitmap of the compressed data on disk. (Hyperion Essbase also uses a 72-byte block header for each block, whether the block is compressed or not.)

Data values in uncompressed data block.  
Each value uses 8 bytes.

25	#MISSING	#MISSING	#MISSING
#MISSING	16	7	#MISSING

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

Figure 40-1: Bitmap Data Compression

## Using RLE Data Compression

When using the run-length encoding (RLE) compression scheme, Hyperion Essbase compresses any consecutive, repetitive values—any value that repeats three or more times consecutively, including zero. Hyperion Essbase keeps track of each repeating value and the number of times it is repeated consecutively.

In the example in Figure 40-2, Hyperion Essbase uses 64 bytes to store the data in the fully expanded block, but uses 56 bytes to store the compressed data on disk. (Hyperion Essbase also uses a 72-byte block header for each block, whether the block is compressed or not.)

Data values in uncompressed data block. Each value that is repeated 3 or more times uses 8 bytes plus a 16-byte repetition factor.

25	#MISSING	#MISSING	#MISSING	← Repeating values (grey)
#MISSING	16	7	#MISSING	← This #MISSING is not a repeating value

Number of **bytes** used to store each value in the same data block.

Cell 1:	8 bytes
Cells 2 - 5:	24 bytes
Cell 6:	8 bytes
Cell 7:	8 bytes
Cell 8:	8 bytes
Total bytes used:	56 bytes

Each value that is not repeated or is repeated only twice uses 8 bytes.

Figure 40-2: RLE Data Compression

## Deciding Which Type of Compression to Use

In most cases, bitmap compression conserves disk space more efficiently. However, much depends on the configuration of the data.

The RLE compression type may be preferable under the following circumstances:

- The average block density in the database is very low (three percent or lower).
- The database has many consecutive zero values or any other consecutive, repeating value.

## Deciding When to Disable Data Compression

You may want to disable data compression if blocks have very high density (90% or greater) and few consecutive, repeating data values. Under these conditions, enabling compression consumes resources unnecessarily.

## Changing Data Compression Settings

Data compression is enabled by default. To change this option from Hyperion Essbase Application Manager, select Database > Settings, click the Storage tab, and select an option from the Data Compression list box. The options are Bitmap (the default), RLE, and No Compression.



You can use the SETDBSTATEITEM and SETDBSTATE commands in ESSCMD to change data compression settings. See the online *Technical Reference* in the DOCS directory for information about the commands. See Chapter 44, “Performing Interactive and Batch Operations Using ESSCMD” for information about ESSCMD.

When you change the data compression type, Hyperion Essbase uses the new option when it writes out data blocks to the database. Changing the compression type has no effect on blocks already on disk (until they are accessed, updated, and written out again).

See “Specifying Data Compression” on page 41-27 for information on changing data compression settings.

## Checking the Compression Ratio

The compression ratio represents the ratio of the compressed block size (including overhead) to the expanded block size, regardless of the compression type in effect. To check the compression ratio, select Database > Information in Hyperion Essbase Application Manager and click the Statistics tab or issue GETDBSTATS in ESSCMD.

**Note:** The compression ratio can vary widely from block to block.

## Data Compression

Because Hyperion Essbase compresses data blocks as they are written to disk, it is possible for bitmap, RLE, and non-compressed data blocks to coexist in the same data file.

- When a compressed data block is brought into the data cache, Hyperion Essbase expands the block to its full size, regardless of the scheme that was used to compress it.
- When Hyperion Essbase stores a block on disk, Hyperion Essbase treats the block the same whether the block was compressed or non-compressed when it was brought into the data cache. In either case, Hyperion Essbase compresses the block according to the specified compression type (including not compressing the block if No Compression is specified).
- If compression is not enabled, Hyperion Essbase writes out the fully expanded block to disk.

## Storage Allocation

Hyperion Essbase automatically allocates storage for index (ESSxxxxx.IND) and data (ESSxxxxx.PAG) files. By default, Hyperion Essbase uses the volume where the ARBORPATH directory resides and uses all the available space on that volume. If you want to use multiple volumes to store your index and data files or if you want to install Hyperion Essbase on one volume and store files on another volume, use the disk volumes setting.

You use the disk volumes setting to control how much space Hyperion Essbase uses to store index files and data files and on which volumes (drives) Hyperion Essbase stores index files and data files.

With the disk volumes setting, you can specify the name of a disk (or a letter corresponding to a disk drive, such as C). If you do not specify disk volumes setting, Hyperion Essbase automatically stores the data and index files in the database directory (for example, `\ESSBASE\SAMPLE\BASIC`).

Hyperion Essbase creates new data and index files when either of the following occurs:

- The total sizes of all files reach the maximum size specified in the disk volumes setting. By default, the total is the sum of all index and data file sizes. If you specify Index under File Type, the total refers to the sum of all index files on a particular volume. Likewise, if you specify Data under File Type, the total refers to the sum of all data files on a particular volume.

For example, suppose you want to use up to 12 GB for Hyperion Essbase files on volume E, 16 GB on volume F, and 16 GB on volume G. Hyperion Essbase creates a new 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.

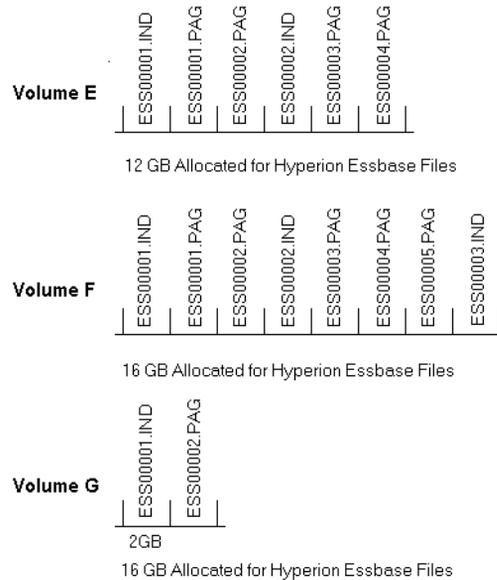
- The size of an individual index or data file on any volume exceeds two gigabytes. In the above example, suppose volumes E and F have reached their capacities and Hyperion Essbase is using volume G. Figure 40-3 illustrates this example.

In volume G, Hyperion Essbase creates file `ESS00001.IND` and fills it to the default limit of 2 GB.

In volume G, Hyperion 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 Hyperion Essbase needs storage when writing index files to disk, Hyperion Essbase creates a new file on

volume G and names it `ESS00002.IND`. Hyperion Essbase then fills `ESS00002.IND` to its 2 GB limit and creates `ESS00003.IND`. Hyperion Essbase follows the same procedures for data files.



*Figure 40-3: Example of How Hyperion Essbase Stores Files Across Volumes*

Hyperion 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`.

You can specify the disk volumes setting from Hyperion Essbase Application Manager menu (select Database > Settings and click the Storage tab).



You can use the `SETDBSTATEITEM` and `SETDBSTATE` commands in `ESSCMD` to specify the disk volumes setting. See the online *Technical Reference* in the `DOCS` directory for information about the commands. See Chapter 44, “Performing Interactive and Batch Operations Using `ESSCMD`” for information about `ESSCMD`.

For more information on how to specify disk volumes and related settings, see “Specifying Disk Volumes” on page 41-20.

## Fragmentation

Even when data compression is enabled, a small amount of fragmentation can occur. Fragmentation is unused disk space. Fragmentation is created when Hyperion Essbase writes a data block to a new location on disk and leaves unused space in the former location of the data block.

The Hyperion Essbase kernel merges adjacent fragments into increasingly larger fragments so that unused space is more likely to be re-used.

## Index and Data File Sizes

You can view index file (.IND file) and data file (.PAG file) names, counts, sizes, and totals and you can determine whether each file is presently open in Hyperion Essbase. From Hyperion Essbase Application Manager, select Database > Information and click the Files tab to view this information.



You can use the LISTFILES command in ESSCMD to view index file (.IND file) and data file (.PAG file) names, counts, sizes, and totals and to determine whether each file is presently open in Hyperion Essbase. See the online *Technical Reference* in the DOCS directory for information about the command.

**Note:** The file size information that is provided by the Windows NT operating system for index and data files that reside on NTFS volumes may not be accurate. The file size information provided by Hyperion Essbase Application Manager and by LISTFILES is accurate.

## Database Restructuring

Usually, as your business changes, your Hyperion Essbase database outline changes. Some database outlines change frequently. Hyperion Essbase has two common ways to change a database outline:

- Using the Outline Editor
- Using dimension building

If changes to your outline are relatively minor, such as renaming a member name, there is little impact to your Hyperion Essbase server. More significant changes, such as moving a member to a different parent, require recalculation of data. The most significant changes are those that affect the way data is stored on the disk. When the data storage arrangement needs to change, you must *restructure* the database. Changes that require restructuring the database are very time-consuming, unless you discard your data before restructuring.

Hyperion Essbase uses three types of restructure operations:

- **Full restructure:** If a member of a dense dimension is moved, deleted, or added, Hyperion Essbase restructures the blocks in the data files and creates new data files. When Hyperion Essbase restructures the data blocks, it regenerates the index automatically so that the index entries point to the new data blocks. Hyperion Essbase marks all restructured blocks as dirty, so you need to recalculate the database after a full restructure. Full restructuring is the most time-consuming of the restructures and, for large databases, can take a relatively long time to complete.
- **Sparse restructure:** If a member of a sparse dimension or a member of an attribute dimension is moved, deleted, or added, Hyperion Essbase restructures the index and creates new index files. Restructuring the index is relatively fast; the amount of time required depends on the size of the index.
- **Outline-only restructure:** Hyperion 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 are using Intelligent Calculation in your database, then any time data blocks are restructured, all restructured blocks are marked as dirty, and the next default Intelligent Calculation is a full calculation. If you change a name or a formula, Hyperion Essbase does not mark the block as dirty. Therefore, you must use a method other than full calculation to recalculate the member or the database.

If you change your outline frequently, you may want to consider enabling incremental restructuring. See “Incremental Restructuring” on page 40-19.

For more information about...	See...
Intelligent Calculation	Chapter 34, “Using Intelligent Calculation to Optimize Calculation”
Sparse and dense dimensions	Chapter 4, “Basic Architectural Elements”
Attribute dimensions	Chapter 10, “Working with Attributes”
Dimension building	Chapter 13, “Introducing Dynamic Dimension Building”
The Outline Editor	Chapter 8, “Creating and Changing Database Outlines”

## The Hyperion Essbase Process for Restructuring Data

When Hyperion Essbase restructures both the data blocks and the index, it uses the following files:

- *dbname*.IND
- ESSxxxxx.IND
- ESSxxxxx.PAG
- *dbname*.OTL
- *dbname*.ESM
- *dbname*.TCT

To perform a full restructure, Hyperion Essbase does the following:

1. Creates temporary files that are copies of the .IND, .PAG, .OTL, .ESM, and .TCT files. The temporary files substitute another character for the last character of the file extension, so the temporary file names are *dbname*.INN, ESSxxxxx.INN, ESSxxxxx.PAN, *dbname*.OTN, *dbname*.ESN, and *dbname*.TCU.

2. Reads the blocks from the old database files, restructures them in memory, and then stores them in the new temporary files. This step takes the most time.
3. Removes the old database files, including .IND, .PAG, .OTL, .ESM, and .TCT files.
4. Renames the temporary files to the correct file names: .IND, .PAG, .OTL, .ESM, and .TCT.

If an error or system failure occurs while Hyperion Essbase is restructuring, it is most likely to occur during step 2. To recover, you can simply restart the database. However, you need to delete the temporary files, both to free up disk space and to avoid conflicts the next time you restructure the database.

If a system failure occurs during steps 1, 3, or 4, then you need to review the directory of your disk and evaluate how far the restructuring has progressed. Recovering involves removing the temporary files, or, if all but step 4 is complete, renaming the temporary files to the correct file names.

When Hyperion Essbase does a sparse restructure (restructures just the index), it uses the following files:

- ESSxxxxx.IND
- *dbname*.OTL
- *dbname*.ESM

To perform a sparse restructure, Hyperion Essbase does the following:

1. Renames the old *dbame*.ESM file to *dbname*.ESR.
2. Renames the old ESSxxxxx.IND files to ESSxxxxx.INM.
3. Creates new index files (ESSxxxxx.IND) to store index information that is changed by the restructuring operation.
4. Removes the old *dbname*.ESR file and the old index files (ESSxxxxx.INM).

If a system failure occurs during any of these steps, you can recover by restarting the database.

## Optimization of Restructure Operations

If your outline changes frequently, analyze your outline and the types of changes you are making. Remember that changes to sparse dimensions or attribute dimensions are relatively fast because only the index needs to change. Changes to dense dimensions are relatively slow because the data blocks need to be rebuilt.

The following are the types of restructure operations, in order of decreasing performance:

- Outline only (no index or data files)
- Sparse (only index files)
- Full (index files and data files) due to added, deleted, or moved members and other operations as listed in Table 40-2
- Full (index and data files) due to a dense dimension becoming sparse or a sparse dimension becoming dense

The following operations are slow:

- Changing a dense dimension to a sparse dimension
- Changing a sparse dimension to a dense dimension

## Actions That Improve Performance

To improve performance during restructure, consider doing the following:

- Restructure only input-level or level 0 data. Restructuring only input-level and level 0 data requires less disk space than restructuring all data. When you change the outline such that Hyperion Essbase needs to restructure data, Hyperion Essbase displays a dialog box that you can use to choose to restructure only input-level or level 0 data. See “Saving a Modified Outline” on page 40-27.
- If you change some dimensions frequently, make them sparse.

## Incremental Restructuring

If you make frequent changes to your outline, you may want to consider enabling incremental restructuring. When incremental restructuring is enabled, Hyperion Essbase defers restructuring so that a change to the database outline or to a dimension does not cause structural change. Hyperion Essbase restructures the index and, if necessary, the affected block the next time the block is accessed.

When incremental restructuring is enabled, Hyperion Essbase defers restructuring for many database changes. See Table 40-2.

The following changes override incremental restructuring; that is, they result in immediate restructuring, regardless of whether incremental restructuring is enabled:

- Adding or deleting a non-attribute dimension
- Deleting a stored member of a sparse dimension
- Changing a dimension definition from sparse to dense or from dense to sparse

Certain member additions and certain changes to sparse dimensions can also trigger immediate restructuring. For more information, see Table 40-2.

You can enable incremental restructuring for any of the following:

- An individual database in an application
- All databases in an application
- All databases in all applications

To enable incremental restructuring, use the `INCRESTRUC` parameter in the `ESSBASE.CFG` file. For more information on the `INCRESTRUC` parameter and syntax, see the online *Technical Reference* in the `DOCS` directory.

Hyperion Essbase logs outline changes in an internal file, `database_name.OCL`. Hyperion Essbase clears the file when it does a full database restructure or when you clear or reset the database. The file `database_name.OCL` can grow quite large in the meantime. To clear this file, issue `VALIDATE` in `ESSCMD`. `VALIDATE` causes Hyperion Essbase to restructure any blocks whose restructure was deferred; thus, the file is cleared. When you issue `VALIDATE`, make sure that

the database is *not* in read-only mode (read-only mode is used for archiving). For more information on the VALIDATE command, see “Checking Structural Integrity” on page 42-13.

**Note:** If you are using linked reporting objects (LROs) in a database, incremental restructuring is automatically disabled on that database. When you enable incremental restructuring for all databases, the presence of an LRO in a database disables INCRESTRUC for that database, but does not affect the other databases on the server.

## Restructuring Considerations

This section discusses features and options that cause Hyperion Essbase to track additional information and that, therefore, affect performance.

### Outline Change Log

If you activate the outline change log, Hyperion 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 Hyperion Essbase must make to the log. By default, Hyperion Essbase does *not* log outline changes. To activate the log, set OUTLINECHANGELOG to True in the ESSBASE.CFG file. For more information about the log, see “Outline Change Log File” on page 46-35.

### IBM Relational Storage Manager

If you use the relational storage manager of the IBM DB2 OLAP Server, any changes to metadata are reflected in both Hyperion Essbase and the DB2 OLAP Server. *Metadata* is data contained in the database outline that describes the values within a database. Dimension and member names are metadata. Data that describes values is stored only on the DB2 OLAP Server.

## Hyperion Essbase Partitioning Option

When you use Hyperion Essbase Partitioning, Hyperion Essbase tracks outline changes so that you can synchronize the database outlines across partitions. Tracking outline changes slows restructuring performance, particularly when there are many structural changes.

If Hyperion 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. Use the Validate page of the Partition Wizard. See “Validating the Partition” on page 16-38 for more information.

**Note:** To validate a partition, you must have DB Designer or higher privileges.

2. Synchronize the outlines of the partitions. Select Database > Synchronize Outline to open the Synchronize Outline dialog box. See “Synchronizing Outlines” on page 16-50 for more information.



You can use ESSCMD to synchronize outlines. See the online *Technical Reference* in the DOCS directory for information about these commands. See Chapter 44, “Performing Interactive and Batch Operations Using ESSCMD” for information about ESSCMD.

## Impact of Common Outline Changes

The following table shows all operations that affect data values and restructuring. It is a good idea to recalculate the database after any type of restructure operation.

The third column in the table refers to incremental restructuring. See “Incremental Restructuring” on page 40-19 if the term is unfamiliar to you.

For information about dynamic calculation members, see Chapter 29, “Dynamically Calculating Data Values.” For information about attribute dimensions, see “About Attributes” on page 10-2.

**Note:** If you are using Hyperion Essbase Partitioning Option, restructuring affects only the local, target database, not the database to which you are connected.

*Table 40-2: How Actions Affect Databases and Restructuring*

<b>Action</b>	<b>Data and Restructure Implications</b>	<b>Incremental Restructuring Implications (If Enabled)</b>
Rename member	None	None
Change member formula	Data needs to be recalculated to reflect changes in formulas.  No restructure.	None
Delete member of sparse dimension	Data needs to be recalculated to reflect changes to relationships.  Hyperion Essbase deletes from the index file pointers that point to blocks represented by the deleted member. Because the blocks are no longer pointed to, they become free space.	Hyperion Essbase restructures the index (overrides incremental restructure).  For shared and label-only members: restructure deferred.
Delete member of attribute dimension	None	None
Delete member of dense dimension	Data needs to be recalculated to reflect changes to relationships.  Hyperion Essbase restructures the data files to reflect a changed block size. Hyperion Essbase restructures the index.	Restructure deferred.

Table 40-2: How Actions Affect Databases and Restructuring (Continued)

<b>Action</b>	<b>Data and Restructure Implications</b>	<b>Incremental Restructuring Implications (If Enabled)</b>
Delete member with shared property in sparse or dense dimension	Data needs to be recalculated. The data remains associated with the original member name, but because the parent of the shared member may have depended on the child data, recalculation is needed.  No restructure.	None
Add member to sparse dimension	Data for the new member needs to be loaded or calculated to derive new values.  Hyperion Essbase restructures the index.	Restructure deferred.
Add member to dense dimension	Data for the new member needs to be loaded or calculated to derive new values. Data needs to be recalculated.  Hyperion Essbase restructures the data files to reflect a changed block size. Hyperion Essbase restructures the index.	Restructure deferred.
Add member to attribute dimension	None	None
Add member with shared property to sparse or dense dimension	Data needs to be recalculated. The new shared member affects the consolidation to its parent.  No restructure.	None

Table 40-2: How Actions Affect Databases and Restructuring (Continued)

<b>Action</b>	<b>Data and Restructure Implications</b>	<b>Incremental Restructuring Implications (If Enabled)</b>
Redefine Dynamic Calc member as Dynamic Calc And Store	For dense dimension members: Hyperion Essbase restructures both index and data files.  For sparse dimension members: No restructure.	Restructure deferred.
Redefine Dynamic Calc And Store member as Dynamic Calc	None	None
Redefine regular dense dimension member as Dynamic Calc And Store	None	None
Redefine regular dense dimension member as Dynamic Calc	Hyperion Essbase restructures both index and data files.	Restructure deferred.
Redefine sparse dimension Dynamic Calc And Store or Dynamic Calc member as regular member	No restructure.	No restructure.
Redefine sparse dimension regular member as Dynamic Calc or Dynamic Calc And Store	Hyperion Essbase restructures both index and data files.	Restructure deferred.
Redefine dense dimension Dynamic Calc And Store member as regular member	No restructure.	No restructure.
Redefine dense dimension Dynamic Calc member as regular member	Hyperion Essbase restructures both index and data files.	Restructure deferred.
Redefine dense dimension regular member as Dynamic Calc member	Hyperion Essbase restructures both index and data files.	Restructure deferred.

Table 40-2: How Actions Affect Databases and Restructuring (Continued)

Action	Data and Restructure Implications	Incremental Restructuring Implications (If Enabled)
Add, delete, or move sparse dimension Dynamic Calc member	Hyperion Essbase restructures only index files.	For member add or delete: restructure deferred.  For member move: Hyperion Essbase restructures only index files (overrides incremental restructure).
Add, delete, or move <i>sparse dimension</i> Dynamic Calc And Store member	Hyperion Essbase restructures only index files.	For member add: restructure deferred.  For member move or delete: Hyperion Essbase restructures only index files (overrides incremental restructure).
Add, delete, or move a dense dimension Dynamic Calc And Store member	Hyperion Essbase restructures both index and data files.	Restructure deferred.
Add, delete, or move a dense dimension Dynamic Calc member	No restructure.	No restructure.
Move regular member within a sparse dimension	Data needs to be recalculated to reflect changes in consolidation.  Hyperion Essbase restructures the index file.	Hyperion Essbase restructures the index file (overrides incremental restructure).
Move regular member within a dense dimension	Data needs to be recalculated to reflect changes in consolidation.  Hyperion Essbase restructures both index and data files.	Restructure deferred.
Move an attribute dimension member	None	None

Table 40-2: How Actions Affect Databases and Restructuring (Continued)

<b>Action</b>	<b>Data and Restructure Implications</b>	<b>Incremental Restructuring Implications (If Enabled)</b>
Change dense/sparse setting	Data needs to be recalculated.  Hyperion Essbase restructures both index and data files.	Hyperion Essbase restructures both index and data files (overrides incremental restructure).
Change label only setting	Data needs to be recalculated.  Hyperion Essbase restructures both index and data files.	Restructure deferred.
Change shared member setting	Data needs to be recalculated to reflect the changed data value of the child.  Hyperion 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.	None
Change the order of two sparse dimensions	No calculation or data load impact.  Hyperion Essbase restructures the index.	Hyperion Essbase restructures the index (overrides incremental restructure).
Change the order of two dense dimensions or change the order of a dense and sparse dimension	Data needs to be recalculated.  Hyperion Essbase restructures both index and data files.	Hyperion Essbase restructures both index and data files (overrides incremental restructure).
Change the order of dimensions	Data needs to be recalculated.  Hyperion Essbase restructures both index and data files.	Hyperion Essbase restructures both index and data files (overrides incremental restructure).
Change the order of attribute dimensions	None	None

Table 40-2: How Actions Affect Databases and Restructuring (Continued)

Action	Data and Restructure Implications	Incremental Restructuring Implications (If Enabled)
Create, delete, clear, rename, or copy an alias table	None	None
Import an alias table or set member alias	None	None
Change the case-sensitive setting	None	None
Name a level and generation	None	None
Create, change, or delete a user-defined attribute	None	None
Change a member alias or add an alias to a member	None	None

## The Hyperion Essbase Process for Restructuring After Outline Changes

This section tells you what to expect when you change, add, or delete dimensions.

### Saving a Modified Outline

In the Outline Editor, if you change a dimension in the outline, such that Hyperion Essbase needs to restructure the data file, Hyperion Essbase displays the following dialog box when you save the outline changes:

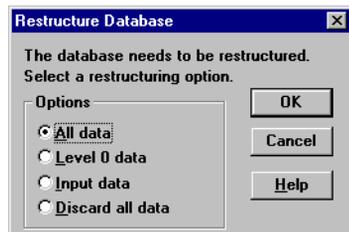


Figure 40-4: Restructure Database Dialog Box

In the Restructure Database dialog box, you specify a restructuring option for blocks. 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.

- All data—This option preserves all data, but is generally slower than the other options.
- Level 0 data—This option preserves data only for level zero (leaf node) members. If you change the database and need to recalculate and if all data required for the calculation is in level zero members, this is the optimal restructure option.

When you select level 0 data, all upper level blocks are deleted before restructuring. This option reduces the disk space required to restructure and improves calculation time when the database is recalculated. Hyperion Essbase recreates the upper-level blocks when you calculate the database.

- Input data—This option preserves only the blocks that contain data that is loaded. Many applications contain data that is entered at parent levels. This option prevents any blocks created by data loading from being deleted, whether they are upper-level or lower-level blocks. If you change the database and need to recalculate and if you load data into various levels of the outline, this is the optimal restructure option.

The Input data option deletes all blocks that contain strictly calculated values before restructuring. This option reduces the disk space required to restructure and improves calculation time when the database is recalculated.

- Discard all data—This option preserves no data. Use this option when you expect to reload the data or when the outline is so radically changed that no existing data applies.

## **Saving an Outline with One or More New Standard Dimensions**

If you add one or more new standard (non-attribute) dimensions, then any data previously existing in the database must be associated with a member of each new dimension. 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. See “Adding and Deleting Standard Dimensions and Stored Members” on page 3-16 for more information.

If you add one or more new standard dimensions, the following dialog box is displayed:

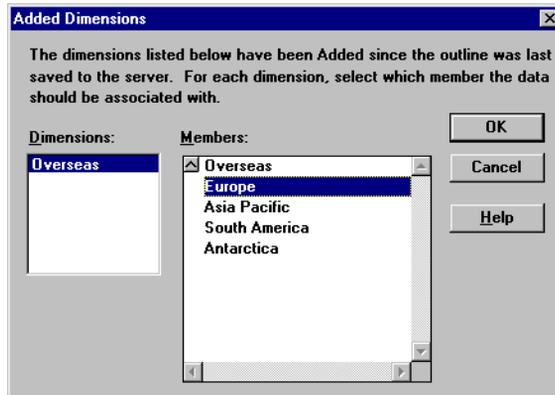


Figure 40-5: Added Dimensions Dialog Box

## Saving an Outline with One or More Deleted Standard Dimensions

If you delete one or more standard (non-attribute) dimensions, then the data associated with only one member of each deleted dimension can be retained. For example, removing a dimension called Market from the outline implies that the data that remains in the database after the restructure operation is associated with a specified member of the Market dimension. See “Adding and Deleting Standard Dimensions and Stored Members” on page 3-16 for more information.

One application of this technique is to delete a dimension from the database, such as Market, save the database as another name, such as New York, and specify the member to keep, such as New York. The new database is a working database for the New York market, and analysts in New York can use it for their analysis.

If you delete one or more dimensions, the following dialog box is displayed:

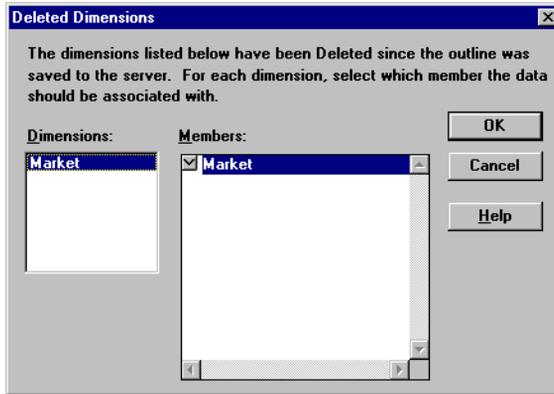


Figure 40-6: Deleted Dimensions Dialog Box

**Note:** Hyperion Essbase may display a “Restructuring not required” message, yet still perform an index-only restructure. This event is most likely to occur if you make changes to a sparse dimension.

If you try to cancel a restructure operation, Hyperion Essbase may issue a “Can’t cancel” message. If such a message is displayed, Hyperion Essbase is performing final cleanup and it is too late to cancel.

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

# Specifying Hyperion Essbase Kernel Settings

This chapter provides a table that lists the major Hyperion Essbase kernel settings and tells you how to change them. It contains the following sections:

- “About Hyperion Essbase Kernel Settings” on page 41-2
- “Specifying and Changing Hyperion Essbase Kernel Settings” on page 41-3
- “Using Hyperion Essbase Application Manager for Database Settings” on page 41-5
- “Using ESSCMD for Database Settings” on page 41-7
- “Specifying Index Cache Size” on page 41-8
- “Specifying Data File Cache Size” on page 41-9
- “Enabling Cache Memory Locking” on page 41-12
- “Specifying Index Page Size” on page 41-14
- “Specifying Isolation Level” on page 41-15
- “Specifying Disk Volumes” on page 41-20
- “Specifying Data Compression” on page 41-27
- “Controlling Data Block Size” on page 41-29

This chapter assumes that one of the following is true:

- You have read Chapter 15, “Estimating Disk and Memory Requirements for a Database” and Chapter 40, “Introducing the Hyperion Essbase Kernel.”
- You are familiar enough with Hyperion Essbase to know the settings values that you want to use.

**Note:** If you migrated from Version 5.x, read the migration information in the *Hyperion Essbase Start Here* booklet before changing the settings described in this chapter.

## About Hyperion Essbase Kernel Settings

You can customize Hyperion Essbase kernel storage settings at the database level—you can have different settings for every database on your server. The information in this section helps you understand what each setting controls and how to adjust each setting. For a table that lists performance-related settings, see Appendix A, “Performance-Related Settings.”

The major Hyperion Essbase kernel settings that you can customize are the following:

- Index Cache Size
- Data File Cache Size
- Data Cache Size
- Cache Memory Locking
- Isolation Level
- Disk Volumes
- Data Compression

This chapter focuses on how to specify these settings. For background information, see Chapter 40, “Introducing the Hyperion Essbase Kernel.”

# Specifying and Changing Hyperion Essbase Kernel Settings

You can specify most Hyperion Essbase kernel settings by using the Database Settings dialog box in Hyperion Essbase Application Manager or by using ESSCMD. In rare cases you may want to use the `ESSBASE.CFG` server file to specify settings.

---

**CAUTION:** In previous versions of Hyperion Essbase you can specify many Hyperion Essbase kernel settings in the `ESSBASE.CFG` server file. In Version 5.x and higher, Hyperion Essbase overrides most of the `.CFG` settings. See “Precedence of Settings” on page 41-4.

---

## ESSCMDs That Control Database Settings

Two ESSCMD commands control database settings: `SETDBSTATE` and `SETDBSTATEITEM`:

- `SETDBSTATE` exists for compatibility with previous versions and uses the syntax `SETDBSTATE parameter value`, in interactive mode; for example, `SETDBSTATE accessLevel 1`.
- `SETDBSTATEITEM` controls more settings than `SETDBSTATE` and uses the syntax `SETDBSTATEITEM n`, in interactive mode. The integer *n* corresponds to a parameter. For example, `SETDBSTATEITEM 4` corresponds to `accessLevel`. If you do not know a parameter number, you can omit it, and Hyperion Essbase lists all parameters and their corresponding numbers. Hyperion Essbase also prompts you for a database and an application name.
- As with other ESSCMD commands, you can supply an application name, a database name, and parameter values on the command line.
- See the online *Technical Reference* in the `DOCS` directory for information about `SETDBSTATE` and `SETDBSTATEITEM` commands. See Chapter 44, “Performing Interactive and Batch Operations Using ESSCMD,” for information about ESSCMD.

## Summary of Database Settings

Hyperion Essbase uses the most recent setting, whether you use SETDBSTATEITEM, SETDBSTATE, or the Database Settings dialog box.

Table 41-1: Summary of Hyperion Essbase Kernel Database Settings

Where Set	When Effective	Overrides
Database Settings dialog box or the SETDBSTATEITEM or SETDBSTATE command in ESSCMD	For index cache, data file cache, data cache, cache memory locking, and disk volumes settings: upon stop and restart of database.	ESSBASE.CFG settings <sup>1</sup>
Scope: Database-specific	For index page size: when database is started but is empty.	SetDbState settings
	For isolation level and concurrency parameters: the next time there are no active transactions.	Hyperion Essbase installation defaults
	For all other settings: immediately.	

## Precedence of Settings

Database settings specified using the Hyperion Essbase Application Manager Database Settings dialog box, or the SETDBSTATE and SETDBSTATEITEM commands in ESSCMD, override ESSBASE.CFG file settings. ESSCMD and dialog boxes provide different ways to change the same database settings. The most recent value of the database setting overrides any previously set value, whether it was set through a command or a dialog box. Both methods permanently override ESSBASE.CFG settings, as the following paragraph explains.

Hyperion Essbase reads the ESSBASE.CFG file when you start the server, and then applies database settings, overriding the ESSBASE.CFG settings. Hyperion Essbase applies a setting from the ESSBASE.CFG file only if you have deleted the database settings. Hyperion Essbase takes the value of the ESSBASE.CFG file setting and uses it as an initial value for the database setting. If you subsequently stop and restart the server, Hyperion Essbase uses the database setting. Hyperion Essbase ignores any changes you make to the setting in the ESSBASE.CFG file, unless you delete the database setting before starting the server. ESSBASE.CFG file settings apply to all databases on a server.

Irrespective of how you enter the settings, some settings take effect immediately, and others take effect the next time the database is started. Therefore, the settings in effect may not be the settings you most recently entered. For information on when settings take effect, see Table 41-1.

To view the most recently entered settings use GETDBSTATE or select Database > Settings in Hyperion Essbase Application Manager. To view the settings currently in effect, use GETDBINFO or select Database > Information in Hyperion Essbase Application Manager.

For information on stopping and starting servers, applications, and databases, see Chapter 45, “Running Hyperion Essbase, Applications, and Databases.”

## Using Hyperion Essbase Application Manager for Database Settings

You can set database, application, and server options by using the settings dialog boxes in Hyperion Essbase Application Manager. This section covers database settings. See Chapter 17, “Managing Security at Global and User Levels” for information on application and server settings.

To check or customize database settings, select Database > Settings in Hyperion Essbase Application Manager.

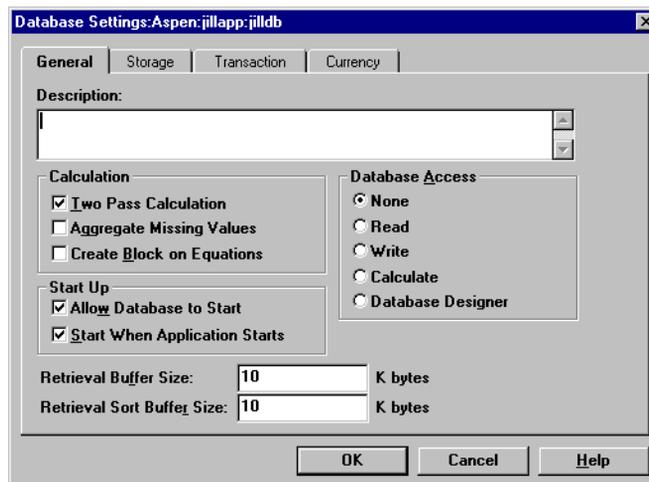


Figure 41-1: The Database Settings Dialog Box

The dialog box has four tabs, which correspond to the following pages.

The General page includes the following settings:

- Calculations—see Part V, “Calculating Data” for more information.
- Database access—see Chapter 17, “Managing Security at Global and User Levels” for more information.
- Database startup—see the Hyperion Essbase Application Manager online help for more information.
- Retrieval buffers—see Chapter 38, “Optimizing Your Reports” for more information.

The Storage page includes the following settings:

- Data compression—see “Specifying Data Compression” on page 41-27 for more information.
- Cache memory locking—see “Enabling Cache Memory Locking” on page 41-12 for more information.
- Index cache size—see “Using Hyperion Essbase Application Manager for Database Settings” on page 41-5 for more information.
- Index page size—see “Specifying Index Page Size” on page 41-14 for more information.
- Data file cache size—see “Specifying Data File Cache Size” on page 41-9 for more information.
- Data cache size—see “Specifying Data File Cache Size” on page 41-9 for more information.
- Location of disk volumes—see “Specifying Disk Volumes” on page 41-20 for more information.

The Transaction page includes the following settings:

- Isolation level—see “Specifying Isolation Level” on page 41-15 for more information.
- Synchronization point—see “Specifying Isolation Level” on page 41-15 for more information.
- Concurrency parameters—see “Specifying Isolation Level” on page 41-15 for more information.

The Currency page includes settings for currency databases. See Chapter 43, “Designing and Building Currency Conversion Applications,” for more information.

**Note:** Changes that you make in any pages of the Database Settings dialog box affect only the specified database.

For information about when settings take effect, see Table 41-1.

## Using ESSCMD for Database Settings



You can use the SETDBSTATEITEM or SETDBSTATE command in ESSCMD to set or change database settings. See the online *Technical Reference* in the DOCS directory for information about the commands. See Chapter 44, “Performing Interactive and Batch Operations Using ESSCMD” for information about ESSCMD.

**Note:** Database settings that you set or change with SETDBSTATEITEM or SETDBSTATE affect only the specified database.

You can include SETDBSTATEITEM (or SETDBSTATE) in batch scripts. For information about writing batch scripts, see Chapter 44, “Performing Interactive and Batch Operations Using ESSCMD.”

For simple items, specify the command, item number, application, database, and value:

```
SETDBSTATEITEM 2 "JILLAPP" "JILLDB" "Y";
```

For items that require multiple parameters, such as Isolation Level (item 18), specify multiple parameters:

```
SETDBSTATEITEM 18 "SAMPLE" "BASIC" "1" "Y" "-1";
```

Use a separate SETDBSTATEITEM command for each parameter; you cannot string item numbers together on the same line.

## Specifying Index Cache Size

Index cache size is an important setting; it determines the maximum memory that Hyperion Essbase allocates to hold index information. This section tells you how to specify the setting and assumes that you have decided what value you want. See “Sizing Your Index, Data File, and Data Caches” on page 15-15 for information about determining an appropriate index cache size.

Hyperion Essbase displays setting values in bytes (B) in ESSCMD and in kilobytes (K) in the Database Settings dialog box.

## Using the Hyperion Essbase Application Manager

Select Database > Settings and click the Storage tab. Enter the desired value *in kilobytes* in the Index Cache text box.

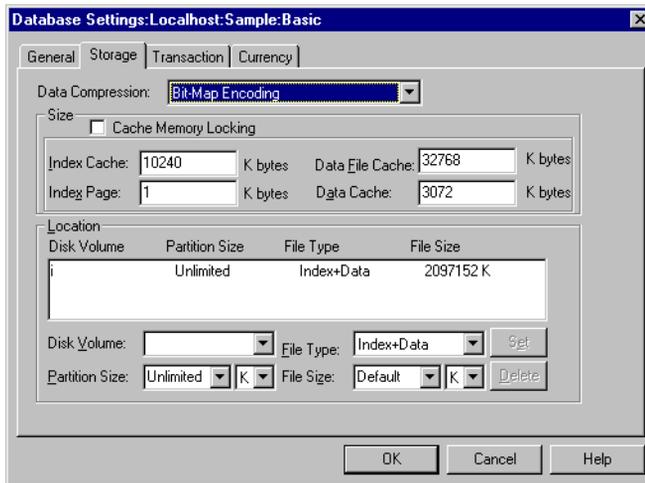


Figure 41-2: Setting Index Cache Size

## Using ESSCMD

Enter `SETDBSTATEITEM 12` or `SETDBSTATE` in ESSCMD and either follow the prompts or supply the required values on the command line, as shown in the following example:

```
SETDBSTATEITEM 12 "JILLAPP" "JILLDB" "10M"
```

When using ESSCMD, you can specify the index cache size in bytes (B), kilobytes (K), megabytes (M), or gigabytes (G). ESSCMD displays minimum, maximum, and current values.

## Making a Change Effective

A change to the index cache size takes effect the next time you start the database.

## Specifying Data File Cache Size

The Data file cache size is an important setting; it determines the maximum memory that Hyperion Essbase allocates to hold data. The data file cache holds data files (.PAG files) in memory.

This section tells you how to specify the setting and assumes that you have decided what values you want. See “Sizing Your Index, Data File, and Data Caches” on page 15-15 for information about determining the appropriate data file cache size.

Hyperion Essbase displays setting values in bytes (B) in ESSCMD and in kilobytes (K) in the Database Settings dialog box.

The minimum value for the data file cache is 8MB (8388608 bytes).

## Using Hyperion Essbase Application Manager

Select Database > Settings and click the Storage tab. Enter the desired value *in kilobytes* in the Data File Cache text box.

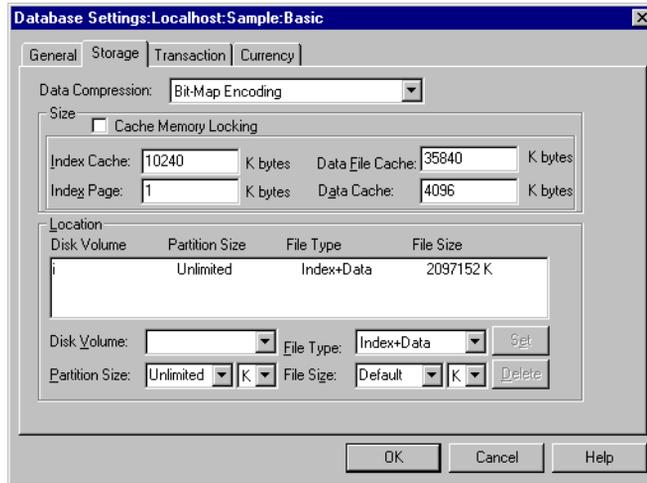


Figure 41-3: Setting Data File Cache Size

## Using ESSCMD

To specify the data file cache, enter SETDBSTATEITEM 27 in ESSCMD and supply the required values on the command line, as shown in the following example:

```
SETDBSTATEITEM 27 "JILLAPP" "JILLDB" "35M"
```

When using ESSCMD, you can specify the data file cache size in bytes (B), kilobytes (K), megabytes (M), or gigabytes (G). ESSCMD displays minimum, maximum, and current values.

## Making Changes Effective

Changes to the size of the data file cache take effect the next time you start the database.

## Specifying Data Cache Size

The data cache holds uncompressed data blocks in memory. In general, if you have to choose whether to allocate memory to the data file cache or to the data cache, choose the data file cache.

This section tells you how to specify the setting and assumes that you have decided what value you want. See “Sizing Your Index, Data File, and Data Caches” on page 15-15 for information about determining the appropriate data cache size.

Hyperion Essbase displays setting values in bytes (B) in ESSCMD and in kilobytes (K) in the Database Settings dialog box.

The minimum value for the data cache is 3MB (3145728 bytes). For compatibility with previous versions of Hyperion Essbase, the Hyperion Essbase kernel accepts a value of 3,000,000 for the data cache and automatically converts the value to the actual minimum of 3MB.

## Using Hyperion Essbase Application Manager

Select Database > Settings and click the Storage tab. Enter the desired value *in kilobytes* in the Data Cache text box.

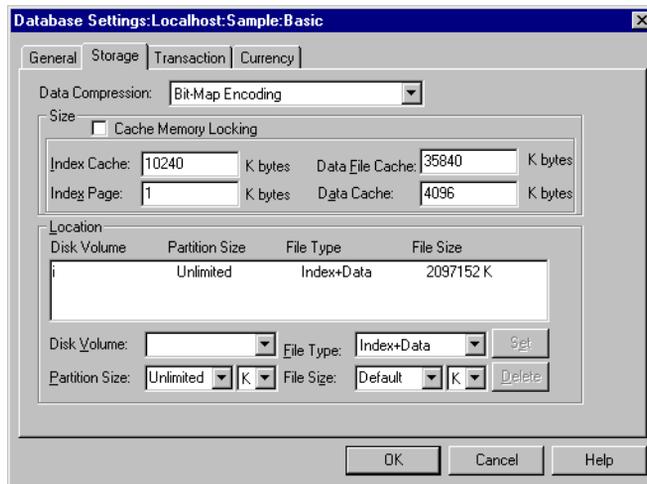


Figure 41-4: Setting Data Cache Size

## Using ESSCMD

To specify the data cache, enter `SETDBSTATEITEM 5` or `SETDBSTATE` in `ESSCMD` and either follow the prompts or supply the required values on the command line, as shown in the following example:

```
SETDBSTATEITEM 5 "JILLAPP" "JILLDB" "4M"
```

To specify the data file cache, enter `SETDBSTATEITEM 27` in `ESSCMD` and supply the required values on the command line, as shown in the following example:

```
SETDBSTATEITEM 27 "JILLAPP" "JILLDB" "35M"
```

When using `ESSCMD`, you can specify the data file cache size and the data cache size in bytes (B), kilobytes (K), megabytes (M), or gigabytes (G). `ESSCMD` displays minimum, maximum, and current values.

## Making Changes Effective

Changes to the size of the data cache take effect the next time you start the database.

## Enabling Cache Memory Locking

The cache memory locking setting locks the memory used for the index cache, the data file cache, and the data cache into physical memory. Locking improves performance for a Hyperion Essbase database because the system memory manager does not need to swap the memory used by the caches when swapping the Hyperion Essbase server process. By default, cache memory locking is turned off.

## Using Hyperion Essbase Application Manager

Select Database > Settings and click the Storage tab. Select the Cache Memory Locking check box.

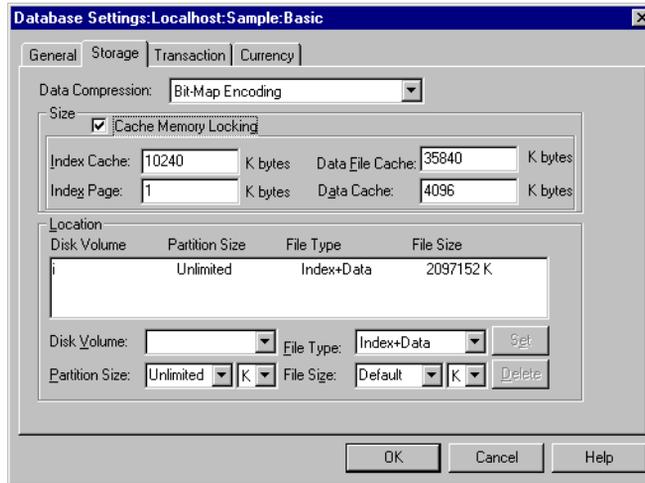


Figure 41-5: Enabling the Cache Memory Locking Setting

## Using ESSCMD

Enter `SETDBSTATEITEM 26` and supply the required values on the command line, as shown in the following example:

```
SETDBSTATEITEM 26 "JILLAPP" "JILLDB" Y
```

## Making a Change Effective

A change to the cache memory locking setting takes effect the next time you start the database.

## Specifying Index Page Size

The index page holds index entries; an index entry exists for every combination of sparse dimensions.

If direct I/O is in effect, Hyperion Essbase ignores the index page size setting and uses a setting of 8K (8,192 bytes). If buffered I/O is in effect, Hyperion Essbase uses the value that is specified in the Index Page Size text box. In this case, the default setting is 1K (1,024 bytes).

## Using Hyperion Essbase Application Manager

Select Database > Settings and click the Storage tab. Enter the desired value *in kilobytes* in the Index Page text box.

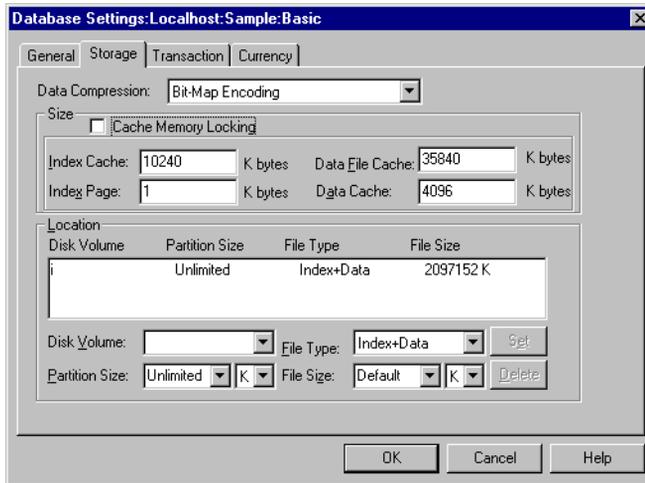


Figure 41-6: Setting Index Page Size

## Using ESSCMD

Enter `SETDBSTATEITEM 13` or `SETDBSTATE` in `ESSCMD` and either follow the prompts or supply the required values on the command line, as shown in the following example:

```
SETDBSTATEITEM 13 "JILLAPP" "JILLDB" "1024"
```

When using `ESSCMD`, you can specify the index page size in bytes (B) or kilobytes (K). The default unit is bytes. `ESSCMD` displays minimum, maximum, and current values.

## Making a Change Effective

A change to the index page setting takes effect the next time you start the database, provided that the database is empty. The database must be empty for the change to take effect.

## Specifying Isolation Level

The Hyperion Essbase kernel offers two levels of isolation for transactions: committed access and uncommitted access. Uncommitted access is the default.

Committed access restricts read/write access on data blocks to one transaction at a time; Hyperion Essbase retains write locks on all affected blocks until the transaction commits. Uncommitted access can produce unpredictable results, because Hyperion Essbase acquires and releases write locks as needed during the transaction. Committed access is recommended. For more information on these isolation levels, see Chapter 42, “Ensuring Data Integrity.”

Depending on the type of access that you specify, Hyperion Essbase uses additional parameters, which you can set in Hyperion Essbase Application Manager or in `ESSCMD`.

## Notes on Isolation Level Options

If you choose committed access, Hyperion Essbase uses these parameters:

- 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 (dialog box) or time out (ESSCMD).
  - 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, Hyperion Essbase displays a lock time out message, and the transaction aborts.
  - A number of seconds that you specify: Your 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, Hyperion Essbase displays a lock time out message, and the transaction aborts.

---

**CAUTION:** Under committed access, Hyperion Essbase retains redundant data until a transaction commits. Allow disk space for double the size of the database to accommodate redundant data.

---

If you choose uncommitted access, Hyperion Essbase uses these 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 non-zero 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.

---

**CAUTION:** Hyperion 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.

---

## Using Hyperion Essbase Application Manager

Select Database > Settings and click the Transaction tab. Click Committed or Uncommitted in the Isolation Level group.

Depending on the type of access that you specify, the Concurrency Parameters group or the Synchronization Point group is available.

Suppose you want committed access, pre-image access, and a 60-second wait for locked data blocks. Specify settings as in the following example:

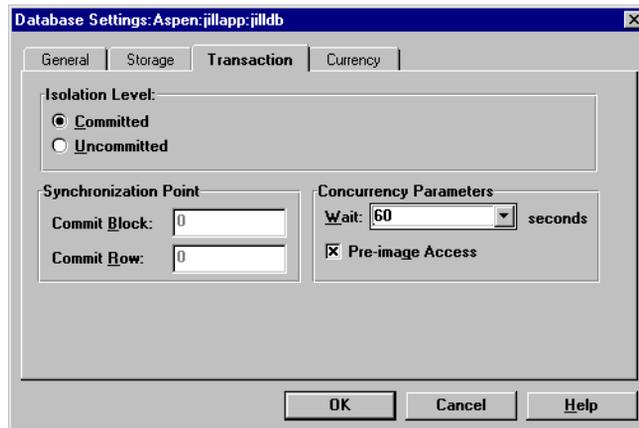


Figure 41-7: Example of Setting Committed Isolation Level

Suppose that you want uncommitted access and that you want Hyperion Essbase to perform a commit operation after 10 blocks are updated, or after 10 rows are loaded. Specify settings as in the following example:

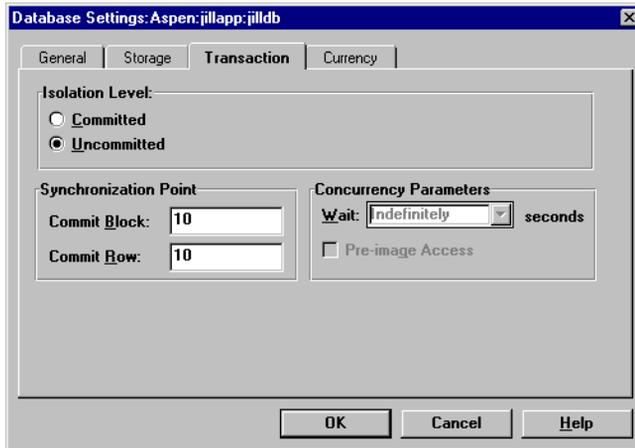


Figure 41-8: Example of Setting Uncommitted Isolation Level

## 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 the type of access that 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 time out (in ESSCMD): -1, 0, or *n*.
  - -1 is indefinite wait.
  - 0 is immediate access, or no wait.
  - *n* is a number of seconds that you specify.

If you choose 2 (uncommitted access), ESSCMD prompts for the following values. See “Notes on Isolation Level Options” on page 41-16 for important details about 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 "JILLAPP" "JILLDB" "1" "Y" "-1"
```

For more syntax information, see the online *Technical Reference* in the DOCS directory.

## Making Changes Effective

Changes to the isolation level settings take effect the next time there are no active transactions.

## Specifying Disk Volumes

Use disk volumes to specify where you want to store Hyperion Essbase index files (`ESSxxxxx.IND`) and data files (`ESSxxxxx.PAG`). See “Storage Allocation” on page 40-11 for information about how Hyperion Essbase allocates storage across volumes.

When you use disk volumes, Hyperion Essbase provides you the following options for each volume:

- Name of the volume
- Maximum space to use on the volume (called Partition Size in Hyperion Essbase Application Manager and Volume Size in `ESSCMD`)
- File type. You can specify index files, data files, or both index and data files. The default is index and data files on the same volume.
- Maximum file size. The default is 2,097,152 kilobytes (two gigabytes); the default is the recommended value. When Hyperion Essbase reaches the maximum file size, it creates a new file and names it incrementally. For example, when `ESS00001.IND` is filled to maximum size, Hyperion Essbase creates `ESS00002.IND`.

Specify the disk volumes in the order in which you want the volumes to be used. You do not have to specify the volume in which Hyperion Essbase is installed as one of the volumes; you can install on one volume and store data on other volumes.

You can tell Hyperion Essbase to stop storing files on a particular volume. Hyperion 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, as shown in Figure 41-9, and click Delete.

**Note:** If you do not specify the disk volumes setting, Hyperion Essbase stores data only on the volume where the `ARBORPATH` directory resides. If the `ARBORPATH` variable is not set, Hyperion Essbase stores data only on the volume where the server was started. For information on setting `ARBORPATH`, see the *Hyperion Essbase Installation Notes*.

If a volume reaches capacity, Hyperion Essbase moves on to the next volume.

If all specified volumes reach capacity, Hyperion Essbase stops any ongoing database operations, issues an error message, and performs fatal error handling. For more information, see “Fatal Error Handling” on page 40-7. If these events occur, shut down the database, allocate more disk space, and restart the database.

You set disk volumes on a per database basis. It is possible for more than one database to 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 Hyperion Essbase files on Volume A.

---

**CAUTION:** If you specify a volume name but not a volume size, Hyperion Essbase uses all available space on the volume.

---

For more information, see “Storage Allocation” on page 40-11.

## Using Hyperion Essbase Application Manager

Select Database > Settings > Storage.

- To add a new volume or to change values for a previously allocated volume:
  1. In the Disk Volume text box, enter the volume name.
  2. Specify Partition Size. Partition Size represents the maximum space that Hyperion Essbase uses on the volume. The default value is `Unlimited`; the minimum is 8,192 kilobytes.
  3. Specify File Type.

You can specify index files, data files, or both. The default is `Index + Data`.
  4. Specify File Size. File Size represents the maximum size that a file specified in File Type can attain before Hyperion Essbase creates a new file. The default value is 2 GB (2097152 kilobytes); the minimum value is 8,192 kilobytes.
  5. Click Set.

**To stop Hyperion Essbase from storing additional files on an allocated volume** (that is, to “deallocate” a volume), enter the volume name in the Disk Volume text box and click Delete. Hyperion Essbase writes no new files to that volume, but continues to access and use files previously created on that volume.

For example, suppose you do not want any more information to be stored on Volume F. Select the volume definition for volume F and press Delete, as shown in Figure 41-9.

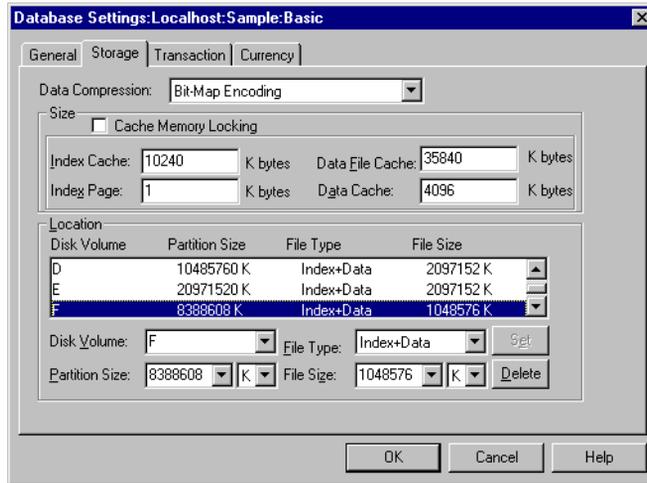


Figure 41-9: Deallocating a Volume

Now, Hyperion Essbase stores no additional data on Volume F.

## Allocating Storage: An Example

Suppose you want to use up to 20 GB for Hyperion 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.

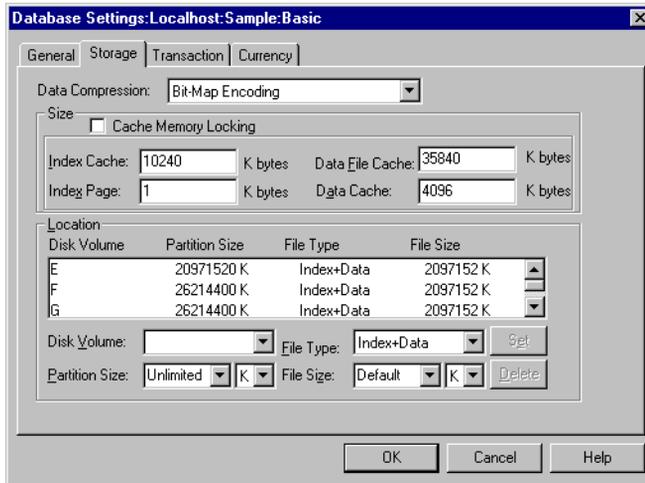


Figure 41-10: Example of Using Disk Volumes

When you load data, Hyperion Essbase stores up to 20 GB on Volume E; if the database is larger than 20 GB, Hyperion Essbase stores the next 25 GB on Volume F, and so on.

## Using ESSCMD

To allocate a new volume, enter `SETDBSTATEITEM 23` 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:

- The number of new disk volumes you want to add

Then, for each new volume, ESSCMD prompts you for:

- Volume name (for each volume)
- Volume size (maximum space to use on the volume)—The default value is Unlimited; the minimum setting is 8 megabytes.

When you use ESSCMD, you can specify volume size in bytes (B), kilobytes (K), megabytes (M), or gigabytes (G). 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 Hyperion Essbase creates a new file)—The default value is 2 gigabytes; the minimum setting is 8 megabytes.

When you use ESSCMD, you can specify file size in bytes (B), kilobytes (K), megabytes (M), or gigabytes (G). ESSCMD displays minimum, maximum, and current values.

The following example allocates up to 10 gigabytes on Volume E, sets a maximum file size of 2 gigabytes, and specifies that data files should be stored only on E:

```
SETDBSTATEITEM 23 "JILLAPP" "JILLDB" "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 gigabytes on Volume C and sets a maximum file size of 2 gigabytes:

```
SETDBSTATEITEM 24 "JILLAPP" "JILLDB" "1" "C" "20G" "3" "2G"
```

**To stop Hyperion 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. Hyperion 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 "JILLAPP" "JILLDB" "4"
```

For more syntax information, see the online *Technical Reference* in the `DOCS` directory.

On UNIX, *volume\_name* is a mounted UNIX file system. You must enter a fully qualified path name up to the name of the directory you are using for Hyperion Essbase. Hyperion Essbase automatically appends the `\app` directory to the path; you do not specify the `\app` directory. Consider the following examples:

```
/vol2/essbase 10M
```

Volume size is the maximum space, in kilobytes, allocated to the volume. The default value is unlimited—Hyperion Essbase uses all available space on that volume.

## Making a Change Effective

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 Data Compression

Hyperion Essbase uses the data compression parameter to specify whether it compresses data on disk and to specify which, if any, compression scheme it uses. See Chapter 40, “Introducing the Hyperion Essbase Kernel,” for information about data compression.

Hyperion Essbase uses the following data compression schemes.

- Bitmap compression, the default. Hyperion Essbase stores only non-missing values and uses a bitmapping scheme.
- Run-length encoding (RLE). Hyperion Essbase compresses repetitive, consecutive values, including zeros and #MISSING values.

You can also disable compression. Compression is enabled by default.

## Using Hyperion Essbase Application Manager

To specify data compression, select Database > Settings and click the Storage tab. Select a value from the Data Compression list box.

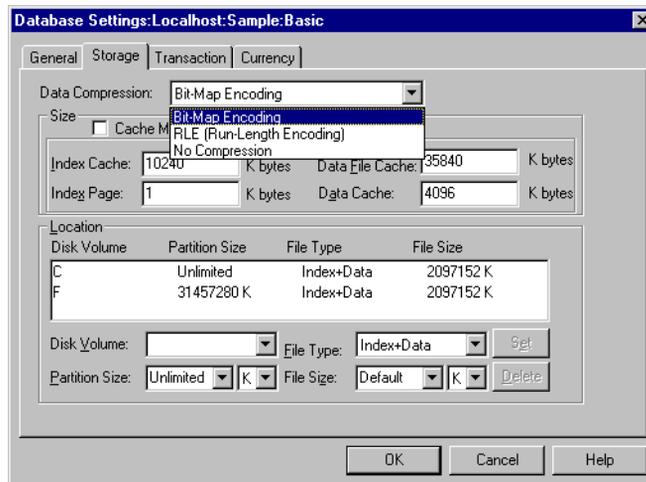


Figure 41-11: Setting Data Compression

## Using ESSCMD

You can enable or disable data compression by using `SETDBSTATEITEM` or `SETDBSTATE`. To set the data compression type, you must use `SETDBSTATEITEM`.

### 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 "JILLAPP" "JILLDB" "Y" "2"
```

For more syntax information, see the online *Technical Reference* in the `DOCS` directory.

### Using SETDBSTATE

Enter `SETDBSTATE` in `ESSCMD`. Follow the prompts until the last prompt, Data Compression on Disk? Enter `Y` (Yes, the default) or `N` (No).

To set the data compression type, you must use `SETDBSTATEITEM` or the Database Settings dialog box.

For more syntax information, see the online *Technical Reference* in the `DOCS` directory.

## Making a Change Effective

Changes to the data compression setting take effect immediately as Hyperion Essbase writes data blocks to disk. For blocks already on disk, Hyperion Essbase does not change compression schemes or enable or disable compression. When you change the data compression settings of blocks already on disk, Hyperion Essbase uses the new compression scheme the next time Hyperion Essbase accesses, updates, and stores the blocks.

## Controlling 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  $8n$  bytes, where  $n$  is the number of cells that exist for that combination of dense dimensions.

**Note:** Eight to 64 kilobytes is the optimum size range.



# Chapter 42

## Ensuring Data Integrity

---

This chapter tells you about how Hyperion Essbase handles transactions and locking and about other ways that Hyperion Essbase protects data.

This chapter includes the following sections:

- “About Transactions” on page 42-1
- “About Isolation Levels” on page 42-2
- “Handling a Transaction” on page 42-4
- “Locking Data” on page 42-5
- “Committing Data” on page 42-9
- “Accommodating Data Redundancy” on page 42-11
- “Rolling Back Transactions” on page 42-11
- “Checking Structural Integrity” on page 42-13

If you are migrating your data from a previous release of Hyperion Essbase, see the *Hyperion Essbase Start Here* booklet.

### About Transactions

When a database is in read/write mode, Hyperion Essbase considers every update request (such as a data load, a calculation, or a statement in a calc script) as a transaction. Hyperion Essbase tracks information about transactions in a transaction control table.

The transaction control table (TCT) contains an entry for each transaction and tracks the current state of each transaction.

## About Isolation Levels

Hyperion Essbase offers two isolation levels for accessing data: *committed access* and *uncommitted access* (the default). You can optimize data integrity by using committed access.

**Note:** Some of the behavior of the Hyperion Essbase kernel depends on which type of access is specified.

## About Committed Access

Committed access provides a high level of data consistency. Only one transaction at a time can update data blocks. Under committed access, the Hyperion Essbase kernel issues short-term read locks and long-term write locks. Hyperion Essbase provides options that determine when Hyperion Essbase issues locks, as the following table shows.

Option	Description	Where Set
Wait or time-out	You decide whether (and how long) to wait for locked blocks to become available or you decide to perform the operation only if the required blocks are immediately available.	Hyperion Essbase Application Manager: Database > Settings, Transaction page  ESSCMD: SETDBSTATEITEM 20
Pre-image access	You view data (in read-only mode) when blocks are locked by another transaction; you view data as it existed when it was last committed (before the other transaction started).	Hyperion Essbase Application Manager: Database > Settings, Transaction page  ESSCMD: SETDBSTATEITEM 19

**Note:** When you have pre-image access enabled, you are not limited to read-only access to data blocks; if you need write access to locked blocks, your transaction waits for write access or times out, depending on the wait or time-out setting. Your transaction gets immediate write access to data blocks that are not locked by another transaction.

If you do not have pre-image access enabled and if you need read or write access to locked blocks, your transaction waits for write access or times out, depending on the wait or time-out setting.

Under committed access, Hyperion Essbase retains some redundant data to ensure data consistency. To accommodate data redundancy, allow disk space for double the size of your entire database.

To enable committed access, change your access type, or change related options, select Database > Settings and click the Transaction page in Hyperion Essbase Application Manager. For more information, see Chapter 41, “Specifying Hyperion Essbase Kernel Settings.”

## About Uncommitted Access

Uncommitted access uses only short-term write locks, so the outcome of concurrent transactions is less predictable than with committed access. Uncommitted access is consistent with the concurrent activity under Hyperion Essbase Version 4. See “Handling a Transaction” on page 42-4 for more information.

For information about storage settings used with uncommitted access (Commit Blocks and Commit Rows), see “Specifying Isolation Level” on page 41-15.

## Handling a Transaction

Hyperion Essbase tracks transactions from start to finish, swapping data blocks in and out of memory as needed and committing data blocks when the transaction completes. The following list describes how Hyperion Essbase handles a transaction: all list items apply to both committed and uncommitted access.

1. A user or batch program begins an operation.
2. The OLAP engine notifies the Hyperion Essbase kernel that a transaction is to begin.
3. The Hyperion Essbase kernel begins the transaction.
4. The OLAP engine requests data from the Hyperion Essbase kernel.
5. The Hyperion Essbase kernel locates the requested data. It passes the data, and some associated control information, to the OLAP engine. If you are using Hyperion Essbase Spreadsheet Add-in, you see this data displayed on the sheet.
6. If you are using Hyperion Essbase Spreadsheet Add-in, when you modify data, you issue the Send command.
7. The Hyperion 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 Hyperion Essbase kernel about the update, and the Hyperion Essbase kernel updates internal data structures accordingly.
9. Steps 4 through 8 repeat as often as necessary to complete the operation.
10. The transaction ends. If Hyperion Essbase encounters an error during transaction processing, it aborts the transaction. If no errors are encountered, Hyperion Essbase commits the transaction. See “Committing Data” on page 42-9 for details on the differences in commit behavior under committed and uncommitted access.
11. Hyperion 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 as 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, Hyperion Essbase may need to break down a single database operation into multiple synchronization points. See “Committing Data with Uncommitted Access” on page 42-10 for information on commit thresholds.

## Locking Data

Hyperion Essbase issues write (exclusive) locks for blocks that are created, updated, or deleted, and read (shared) locks for blocks that need to be accessed but not modified. By issuing the appropriate locks, Hyperion Essbase ensures that data changed by one operation cannot be corrupted by a concurrent update.

**Note:** The Hyperion Essbase kernel locks data blocks, not objects. For information about locking and unlocking outlines and other objects, see “Locking and Unlocking Objects” on page 47-14.

*Table 42-1: 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&send.
Read (shared) lock	Allows other transactions read-only access to the data block, but prevents other transactions from modifying the data block.

The following table shows the locks that Hyperion Essbase issues for various types of operations.

*Table 42-2: Locking by Higher-Level Functions*

<b>Type of Operation</b>	<b>Lock Issued</b>
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 Hyperion Essbase handles locking depends on whether committed or uncommitted access is enabled. For information on access type, see “About Transactions” on page 42-1.

## Locking Under Uncommitted Access

Under uncommitted access, Hyperion Essbase issues short-term write locks; Hyperion Essbase locks blocks only as it is updating them (contrast with committed access, when Hyperion Essbase holds locks until a transaction completes).

Table 42-3 illustrates locking behavior under uncommitted areas when more than one transaction is contending for a lock on the same data. In the example in Table 42-3, transaction Tx1 is running, and transaction Tx2 is requesting access to the same data.

*Table 42-3: Locking Behavior with Uncommitted Access*

<b>Request made by Tx2</b>	<b>If Tx1 holds read lock</b>	<b>If Tx1 holds write lock</b>
Read lock	Tx2 gets read lock.	Tx2 gets read lock.
Write lock	Tx2 gets write lock.	Tx2 waits for Tx1 to release the lock.

## Locking Under Committed Access

Under committed access, Hyperion Essbase issues short-term read locks and long-term write locks. A transaction locks and holds the lock on each block that it modifies until the transaction completes.

Table 42-4 illustrates locking behavior under committed access when more than one transaction is contending for a lock on the same data. In the example in Table 42-4, 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:

- **Pre-image access**—You view data (in read-only mode) when blocks are locked by another transaction; you view data as it existed when it was last committed (before the other transaction started). Enabling pre-image access does not limit you to read access; if you need write access, or if you do not enable pre-image access, you can use the wait or time-out option. Your transaction can immediately get write access to data blocks that are not locked by another transaction.
- **Wait or time-out**—You specify how long your transaction should wait for locked data blocks to become available. If you choose 0 seconds, Hyperion Essbase issues a time-out error if the blocks are not immediately available to your transaction and aborts your transaction.

For information on setting these options, see Chapter 41, “Specifying Hyperion Essbase Kernel Settings.”

*Table 42-4: Locking Behavior Under Committed Access*

		<b>Tx1 holds read lock; Tx2 requests read lock</b>	<b>Tx1 holds read lock; Tx2 requests write lock</b>	<b>Tx1 holds write lock; Tx2 requests read lock</b>	<b>Tx1 holds write lock; Tx2 requests write lock</b>
Pre-image access enabled	Wait (time-out) 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 (time-out) period specified (immediate time-out)	Tx2 gets read lock.	Hyperion Essbase issues time-out message and aborts the transaction.	Tx2 gets pre-image access.	Hyperion Essbase issues time-out message and aborts the Tx2 transaction.
No pre-image access	Wait (time-out) 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 (time-out) period specified (immediate time-out)	Tx2 gets read lock.	Hyperion Essbase issues time-out message and aborts the Tx2 transaction.	Hyperion Essbase issues time-out message and aborts the Tx2 transaction.	Hyperion Essbase issues time-out message and aborts the Tx2 transaction.

## Concurrency Under Committed Access

Occasionally under committed access, a deadlock situation results when two transactions are locking or waiting for access to the same blocks. 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. Neither transaction can complete under these conditions.

Hyperion Essbase transactions periodically perform deadlock detection. If a transaction is waiting to acquire a lock, it checks periodically. If the transaction detects a deadlock, Hyperion Essbase issues an error message, and the transaction aborts.

If you try to update a block that is locked to another user, Hyperion Essbase behaves as follows:

- If wait is set to indefinite, your transaction waits to acquire the needed locks.
- If wait is set to 0 (immediate) and if the required blocks are not immediately available, Hyperion Essbase displays an error message, and aborts the transaction.
- If wait is set to a user-specified number of seconds and the time has expired, Hyperion Essbase displays an error message and aborts the transaction.
- If your request times out, try the operation again.

For information on setting locking options, see Chapter 41, “Specifying Hyperion Essbase Kernel Settings.”

## Committing Data

Hyperion Essbase automatically commits any kind of update operation—data load, calculation, restructure, or spreadsheet operation—that modifies data blocks.

Hyperion Essbase handles commit operations differently under committed and uncommitted access.

You cannot explicitly commit data under either type of access; there is no COMMIT command.

## Committing Data with Committed Access

Under committed access, as Hyperion Essbase completes an operation, it commits the transaction. There is no explicit commit operation for the user to perform; Hyperion Essbase handles the commit automatically.

## Committing Data with Uncommitted Access

Under uncommitted access, the following conditions cause Hyperion Essbase to commit data during one or more synchronization points during a transaction:

- An operation is completed.
- The number of modified data blocks reaches a user-defined threshold. To set the threshold level, select Database > Settings in Hyperion Essbase Application Manager, click the Transaction tab, and use the Synchronization Point group. For more information, see Chapter 41, “Specifying Hyperion Essbase Kernel Settings.”
- The number of loaded data rows reaches a user-defined threshold. To set the threshold, select Database > Settings in Hyperion Essbase Application Manager, click the Transaction tab, and use the Synchronization Point group. For more information, see Chapter 41, “Specifying Hyperion Essbase Kernel Settings.”

If a user-defined threshold is exceeded during an operation, Hyperion Essbase issues a synchronization point to commit the data processed to that point. Hyperion Essbase performs as many synchronization points as are necessary to complete the operation.

## Accommodating Data Redundancy

To ensure data integrity, the Hyperion Essbase kernel temporarily retains redundant (duplicate) information. To accommodate redundant information, allow disk space for double the size of your database.

Hyperion Essbase maintains a file called *dbname.ESM*, in which it stores crucial control information.

---

**CAUTION:** The *databasename.TCT* file, *databasename.ESM* file, the index files, and the data files contain information crucial for data recovery. Never alter or delete these files.

---

## Rolling Back Transactions

If a server crash or other server interruption occurs while there are active transactions running, the Hyperion Essbase kernel rolls back the transactions when the server is restarted.

### Rollback with Committed Access

Under committed access, if the server crashes, Hyperion Essbase rolls back all database updates by transactions that were in progress when the server stopped. Thus, Hyperion Essbase ensures that changes made by the aborted transactions are undone.

If a transaction is aborted due to a non-fatal error, all changes made by the transaction are rolled back.

## Rollback with Uncommitted Access

Under uncommitted access, if the server crashes, Hyperion Essbase rolls back all database updates from the point of the last successful commit. Some of the 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 due to a non-fatal error, Hyperion Essbase commits only the data that the transaction finished processing prior to the abort of the transaction.

## Actions to Take When a Transaction Does Not Complete

Table 42-5 shows what action to take if a transaction does not complete, because the transaction aborted or the server was interrupted.

*Table 42-5: Actions to Take if a Transaction Stops Before Completing*

<b>Operation</b>	<b>Action to Take</b>
Calculation	Start over.
Calc script	Check the server log and see how far the calc script got and decide whether to start over or continue.
Normal data load	Start over.
Arithmetic data load (loads that perform operations on values in the database, such as adding 10 to each value)	Start where the data load left off.
All other operations	Start over.

## Checking Structural Integrity

The VALIDATE command performs several types of structural 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.
- Restructures data blocks whose restructure was deferred with incremental restructuring. For more information, see “Incremental Restructuring” on page 40-19.
- Checks every block in the database to make sure each value is a valid floating point number.
- Verifies the structural integrity of the linked reporting objects (LROs) free space catalog.

As Hyperion Essbase encounters mismatches, it records error messages in the VALIDATE error log file. You can specify a file name for error logging; Hyperion Essbase prompts you for this information if you do not provide it. The VALIDATE utility continues running until it has checked the entire database.



You can use the VALIDATE command in ESSCMD to perform these structural integrity checks. See the online *Technical Reference* in the DOCS directory for information about the command. See Chapter 44, “Performing Interactive and Batch Operations Using ESSCMD” for information about ESSCMD.

During index free space validation, the VALIDATE command verifies the structural integrity of free space information in the index. If integrity errors exist, Hyperion Essbase records them in the VALIDATE log file. 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 any one of three ways:

- Restoring the database from a recent system backup
- Exporting data from the database; creating a new, empty database; and loading the exported data into the new database
- Restructuring the database

VALIDATE also validates the consistency of data block keys. The Hyperion Essbase index contains a key for every data block. Every corresponding data block contains the same key. The VALIDATE utility uses this key as a basis for comparison. You may want to run VALIDATE periodically to ensure data consistency.

Even if you do not use VALIDATE, Hyperion Essbase automatically performs certain validity checking whenever a read operation is performed, to ensure that the index is properly synchronized with the data.

For every read operation, Hyperion 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 Hyperion Essbase encounters a mismatch, it displays an “Invalid block header” error message.



# Designing and Building Currency Applications

Part VIII describes how to design and build Hyperion Essbase OLAP Server currency applications. Part VIII contains the following chapter:

- Chapter 43, “Designing and Building Currency Conversion Applications,” introduces you to currency applications and describes how to create one by creating the main database outline, creating the currency database outline, and linking the two outlines. It also describes how to calculate currency conversions.



# Designing and Building Currency Conversion Applications

---

Organizations with offices in different countries generally do business in the currency of the host country (known as the *local currency*). World and regional headquarters of such organizations must convert data entered in local currencies to a *common currency* for consolidation and analysis. The Hyperion Essbase Currency Conversion option is designed to meet the needs of this complex business problem. This option can be licensed as an “add-on” to the Hyperion Essbase OLAP Server.

The Currency Conversion option includes a sample currency application that contains two databases: INTERNTL and XCHGRATE. This chapter provides background information, step-by-step instructions for building the sample currency application, and procedures for calculating and reporting currency conversions after you have built your own currency application.

This chapter contains the following topics:

- “Overview of the Business Problem” on page 43-2
- “Overview of the Currency Application’s Structure” on page 43-2
- “Steps for Creating a Currency Conversion Application” on page 43-7
- “Creating the Main Database Outline” on page 43-7
- “Creating the Currency Database Outline” on page 43-17
- “Linking the Main and Currency Databases” on page 43-24
- “Calculating Currency Conversions” on page 43-26

**Note:** Your Hyperion Essbase server installation includes the option to install the Sample databases, INTERNTL and XCHGRATE. If you don’t have access to these databases, contact your Hyperion Essbase administrator. For information about installing sample applications, see the *Hyperion Essbase Installation Notes*.

## Overview of the Business Problem

This chapter focuses on solving the business problems that arise as the sample company, The Beverage Company (TBC), expands its business outside the United States. This chapter builds on the business scenario introduced in Chapter 5, “Designing a Single-Server Application,” as TBC adds the following markets:

- Three locations in Canada: Toronto, Vancouver, and Montreal
- Four locations in Europe: the UK, Germany, France, and Spain

In addition, TBC adds a new member, US, which is a consolidation of data from the United States regions: East, West, South, and Central.

Data for each TBC market location is captured in local currency. Dollar values are derived by applying exchange rates to local values. These values must then be converted to a common currency (in this case, US\$).

TBC needs to 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.

When all actuals have been processed, budget data is converted with budget exchange rates.

## Overview of the Currency Application’s Structure

Currency conversion applications are defined by linking a currency database to a main application database, as illustrated in Figure 43-1. The *Main Database* contains the full database outline and associated data values; the *Currency Database* contains only exchange rates and other currency-related information.

In the example provided in this chapter, TBC's Main Database is INTERNTL; its Currency Database is XCHGRATE. On your server, these databases are in the Sample application.

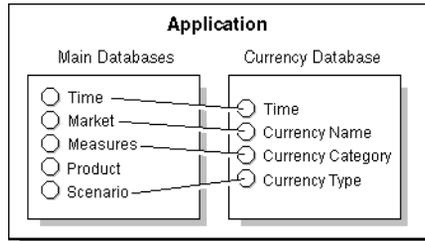


Figure 43-1: Currency Application Consists of Two Databases

## Contents of the Main Database

The Main Database can be from 3 to  $n$  dimensions in size. At a minimum, the database must contain the following dimensions:

- A dimension tagged as Time—In the TBC example this dimension is Year.
- A dimension tagged as Accounts—Within an Accounts dimension, individual members may be tagged with different categories of exchange rates. This allows you to apply different exchange rates to various accounts or measures.

For example, P&L accounts may use exchange rates that differ from those used with balance sheet accounts. In addition, some accounts may not require conversion. For example, members such as Units, Headcount, and Margin% require no conversion.

- A market-related dimension tagged as Country—In a market dimension, each member has an associated currency name. In the TBC example, the currency name US\$ is defined for the Market dimension tagged Country, because all local currencies for the new markets must be converted to US\$, the company's common currency.

Because many 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 43-1, TBC uses only six currency names for its 15 Market dimension members. The children of the member Europe all use a different currency and, therefore, must be assigned individual currency names. However, in the case of the Country dimension and the US member

and its four cities, all locations use the same currency. The same is true of the Canada member and its three cities. When the children of a given member share a single currency, you only need to define a currency name for the parent member.

Table 43-1: TBC Currency Names

Dimension/Member	Currency Name
<b>Market - Country</b> US East West South Central	US\$
<b>Canada</b> Toronto Vancouver Montreal	CN\$
<b>Europe</b>	
<b>UK</b>	Pound
<b>Germany</b>	Mark
<b>France</b>	Franc
<b>Spain</b>	Peseta

A typical Main Database also contains an *optional* fourth dimension: “A dimension defined with a currency partition.” Databases are usually partitioned in a Scenario dimension. Creating a currency partition allows Hyperion Essbase to internally track currency relationships and previously-converted values of the main and exchange rates databases.

**Note:** A currency conversion partition applies only to the Currency Conversion option. It is not related to the Hyperion Essbase Partitioning option that enables data to be shared between databases using a replicated, linked, or transparent partition.

The *Hyperion Essbase Spreadsheet Add-in User's Guide* provides examples of ad hoc currency reporting capabilities. Report scripts also let you define reports that convert data when the report is displayed, as discussed under “Converting Data to a Different Currency in Reports” on page 43-29.

## Contents of the Currency Database

The Currency Database requires the following three dimensions:

- A Currency Time dimension—Exchange rates are usually defined for each time period. 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 are usually converted to the exchange rates of their respective time period from the Currency Database (although you can convert data values against any period's exchange rate).
- Currency Name dimension—This dimension 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.
- Currency Category dimension—This dimension contains all the account categories of the Main Database. The category names define the exchange rate mapping when a conversion occurs. In the TBC example, the account categories included are P&L (Profit & Loss) and B/S (Balance Sheet).

A currency database typically includes an *optional* fourth dimension: “A Currency Type dimension” which contains members that identify various currency scenario types. Typically an application has different exchange rates for scenarios, such as actual, budget, and forecast. Members of the currency-type dimension are not directly mapped to members of the Main Database. Therefore, member names in this dimension are not required to match member names of the Main Database.

## Conversion Methodologies

Different currency applications have different conversion requirements. Hyperion Essbase supports two currency database types, each with a different conversion method:

- **A database containing only converted data values.** Some applications require only converted values to be stored in the Main Database. Local values are entered and the conversion operation overwrites the local values with a common currency. This method assumes that there is no requirement for reporting or analyzing local currencies.

Because this operation overwrites data, you must load local values and recalculate the data each time you perform a conversion. This method is useful only when you want to perform a single (not ongoing) conversion.

- **A database containing both local and converted data.** Most applications require data to be stored in both local and master values. This method permits reporting and analyzing local data. In addition, data modifications and recalculations are easier to control. To use this method, you must define a currency partition (see “Modify the Scenario Dimension” on page 43-14). The Beverage Company (TBC) uses this method for defining their currency application.

Either of these two methods may also require a currency conversion to be applied at report time. This allows you to analyze various exchange rate scenarios without actually storing data in the database. The currency conversion module allows you to perform ad hoc conversions with the Spreadsheet Add-in, which is discussed in the *Hyperion Essbase Spreadsheet Add-in User's Guide*, or with a report script, as discussed under “Converting Data to a Different Currency in Reports” on page 43-29.

# Steps for Creating a Currency Conversion Application

To accommodate the new markets TBC has added and to provide Hyperion Essbase with required currency-related information, TBC has modified their existing Hyperion Essbase database outline and created a currency conversion application. You can use the TBC model to create your own currency conversion application, revising the steps as needed to fit your specific requirements.

The TBC currency conversion application was created using the following steps:

1. “Creating the Main Database Outline” on page 43-7
  - “Open the Existing Database Outline” on page 43-8
  - “Modify the Measures Dimension” on page 43-8
  - “Modify the Market Dimension” on page 43-11
  - “Modify the Scenario Dimension” on page 43-14
  - “Save the Main Database Outline Changes” on page 43-17
2. “Creating the Currency Database Outline” on page 43-17
  - “Generate the Currency Database Outline” on page 43-18
  - “Review the Contents of the Currency Database Outline” on page 43-21
  - “Add New Members to the CurType Dimension” on page 43-22
  - “Save the Currency Database Outline Changes” on page 43-23
3. “Linking the Main and Currency Databases” on page 43-24

## Creating the Main Database Outline

To create the Main Database outline (INTERNTL), you need to open your existing Hyperion Essbase database outline, make changes to its contents, and then save the outline for use in your currency conversion application.

## Open the Existing Database Outline

Begin creating the Main Database outline by opening TBC's existing outline, as follows:

1. Start Hyperion Essbase Application Manager and select Server > Connect.
2. Choose the Hyperion Essbase Server that contains the Sample application.
3. Select Sample from the Applications list box and open the database outline for the database INTERNTL, as shown in Figure 43-2.

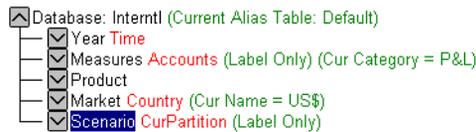


Figure 43-2: Sample Database INTERNTL Hyperion Essbase Outline

TBC has modified the Measures, Market, and Scenario dimensions. The Year and Product dimensions require no changes because they have no information specifically related to currencies.

**Note:** The Year dimension *must* be tagged as the Time dimension.

## Modify the Measures Dimension

To create their currency conversion Main Database, TBC modified the Measures dimension using the following steps. You can follow along and revise the procedures as necessary to create your own currency application.

1. Tag Measures as an Accounts dimension.

You must have a dimension tagged as Accounts in your Main Database. To meet varying conversion and exchange rate requirements, individual members within the Accounts dimension can then be tagged with different categories of exchange rates.

All descendants of the Profit member, for example, use a special currency rate that applies to Profit and Loss accounts. Therefore, they are defined with a currency category of P&L.

2. Define the Profit member with a currency category name, as follows:

- a. Select the Profit member and click the Data Dictionary button, , on the toolbar. The Member Specification dialog box is displayed.
- b. Select the Category option button from the Currency Conversion group, and enter the category name of P&L (for Profit and Loss).

**Note:** Each descendant of a member inherits the currency category tag of its ancestor. A member or sub-branch of members can also have its own category defined.

Because the Measures dimension shows data for Profit, it must also have the P&L currency category name applied.

3. Define the Measures dimension with a currency category name, as follows:

- a. Select Prev to move to the previous member in the dimension and select Category from the Currency Conversion group. The name P&L is displayed from the previous entry.
- b. Click OK to close the dialog box.

All descendants of the Inventory member use a special rate that applies to balance sheet accounts. Define them with a currency category of B/S (for Balance Sheet).

4. Define the Inventory member with a currency category name, as follows:

- a. Select the Inventory member and click the Data Dictionary button, , on the toolbar. The Member Specification dialog box is displayed.
- b. Select Category from the Currency Conversion group, and enter the category name of B/S.
- c. Click OK to close the dialog box.

Ratio members do not require a currency conversion, because the local and converted values yield the same result.

5. Define the Ratio member as not requiring a conversion:

- a. Select the Ratios member, and press the Data Dictionary button, , on the toolbar. The Member Specification dialog box is displayed.
- b. Select No Conversion from the Currency Conversion group.

**Note:** The children of a member defined with No Conversion do not inherit the No Conversion definition. Each member must be defined individually.

6. Define the children of the Ratio member (Margin % and Profit %) as not requiring a conversion:

- a. Select Next to move down the member list to Margin %.
- b. Define Margin % with the No Conversion option.
- c. Select Next to move down the member list to Profit %.
- d. Define Profit % with the No Conversion option.
- e. Click OK to close the dialog box.

Figure 43-3 shows the resulting Measures dimension outline after making the modifications necessary for the TBC currency conversion application.

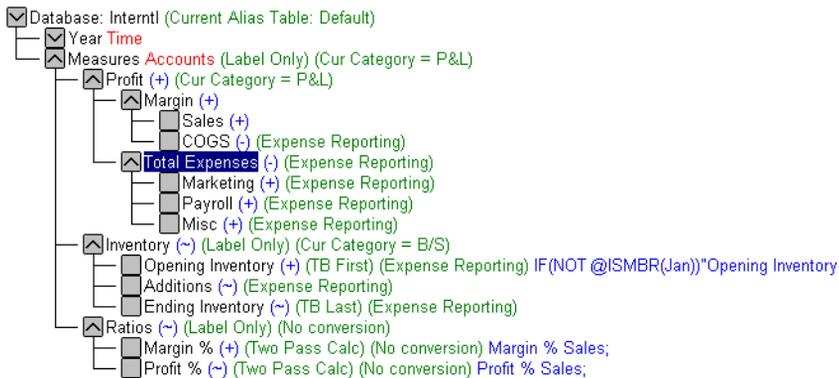


Figure 43-3: TBC Measures Main Database Outline

## Modify the Market Dimension

TBC needed to make the following modifications to the Market dimension to accommodate their newly added markets and create their currency conversion application Main Database. Procedures for making these changes follow the bullet list.

- Tag the Market dimension as Country. Remember, having a Country tag enables you to define the common currency to be used for converting local currency values. In the TBC example, local currency values for their new markets are converted to US\$, the currency name defined for the Country dimension.
- Add the US member, which consolidates data from existing United States region members East, West, South, and Central.
- Add the Canada member and its children, Toronto, Vancouver, and Montreal.
- Add the Europe member and its children, UK, Germany, France, and Spain.
- Define currency names for each of the newly added dimensions/members, as necessary. Remember, when the children of a given member share a single currency, you only need to define a currency name for the parent member.

Modify the Market dimension as follows:

1. Tag Market as a Country dimension by selecting the Market dimension and clicking Country, , on the toolbar.

The existing members (East, West, South, and Central) belong in TBC's newly added US market. To make this outline change, TBC added the US member with the existing members as descendants.

2. Add the new US member:
  - a. Select the Market member and click the Add Child button, .
  - b. Type the name US in the Member Edit box and press Enter twice.
3. Make existing members East, West, South, and Central, children of US by clicking the Add Child button for each member.

**4.** Add Canada and its respective cities to the Market dimension:

- a. Select the US member and click the Add Sibling button, .
- b. Type the name Canada in the edit box and press Enter twice.
- c. Select the Canada member and click the Add Child button.
- d. Add the Toronto, Vancouver, and Montreal members and press Enter twice.

**5.** Add Europe and its respective countries to the Market dimension:

- a. Select the Canada member and press the Add Sibling button.
- b. Type the name Europe in the edit box and press Enter twice. Select the Europe member and click the Add Child button.
- c. Add the UK, Germany, France, and Spain members and press Enter twice.

You must now define currency names for each appropriate member. Because TBC required the database to be converted to US dollars, they defined the Market dimension with the US\$ currency name.

**6.** Define currency names for the Market dimension and its members, as appropriate:

- a. Select the Market dimension and click the Data Dictionary button, . The Member Specification dialog box displays.
- b. Enter US\$ in the Currency Name type in box.
- c. Click OK to close the dialog box.

All descendants of the Market dimension inherit the US\$ currency name (unless a child branch has already been defined with another currency). Because members in the US branch use US\$ as their local currency, there is no need to define a currency name for these members.

Like the Market dimension, all descendants of Canada also use the same currency (CN\$). You need to define a currency name for the Canada member only and its children inherit the CN\$ currency name.

**7. Define a currency name for the Canada member:**

- a.** Select Canada and click the Data Dictionary button, . The Member Specification dialog box is displayed.
- b.** Type CN\$ in the Currency Name box.
- c.** Click OK to close the dialog box.

The children of Europe all use different local currencies. Unlike the Market dimension and the Canada member, you must define each Europe member with an individual currency name.

**8. Define individual currency names for children of the Europe member:**

- a.** Select the UK member name and click the Data Dictionary button, . The Member Specification dialog box is displayed.
- b.** Type Pound in the Currency Name box.
- c.** Click Next to define the currency name for next country (Germany).
- d.** Type Mark in the Currency Name box.
- e.** Click Next to define the currency name for next country (France)
- f.** Type Franc in the Currency Name box.
- g.** Click Next to define the currency name for next country (Spain).
- h.** Type Peseta in the Currency Name box.
- i.** Click OK to close the dialog box.

Figure 43-4 shows the resulting Market dimension outline after making the modifications necessary for the TBC currency conversion application.

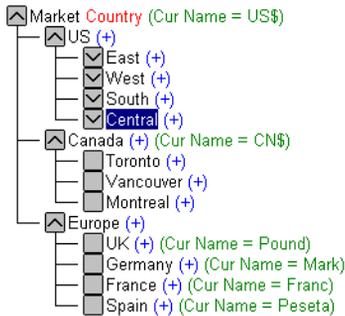


Figure 43-4: TBC Market Dimension Database Outline

## Modify the Scenario Dimension

In the TBC Main Database (INTERNTL), the Scenario dimension contains members for both local and converted values. TBC needed to make the following modifications to the Scenario dimension to create their currency conversion application Main Database:

- Create a currency partition. This enabled TBC to store both converted and non-converted values in their Main Database (INTERNTL).
- Define the converted members under a Dollars parent. This enabled TBC to track actuals converted at actual exchange rates.
- Add a new member called Actual @ Bud XChg. This enabled TBC to track actuals converted at separate budget exchange rates to analyze variances due to exchange rates of their newly added markets.
- Define members that can store data in local currency values (Actual and Budget).
- Apply the Exclude member from consolidation tag to the Local, Bud member, because it doesn't store data.
- Tag the Local member as a label, because it doesn't store data.
- Tag the Scenario dimension as a label, because it also does not store data.

Modify the Scenario dimension as follows:

1. Set a Currency Partition tag on the Scenario dimension:
  - a. Select the Scenario dimension name and click the CurPartition button, .
  - b. Define the converted members under a parent called Dollars.
  - c. Add the member Dollars as a child of Scenario. Select the dimension Scenario and press the Add Child button.
  - d. Type Dollars in the member edit box and press Enter twice.
  - e. Move the Actual, Budget, Variance, and Variance % members as children of Dollars.

The Dollars member does not store data and must, therefore, be tagged as a label.

2. Tag the Dollars member as a label by selecting it and clicking the Label button, .

TBC also wants to track actuals using the budget exchange rate. The Scenario dimension requires a new member to handle this data.

3. Add a new member, Actual @ Bud XChg, to the Scenario dimension Actual member:
  - a. Select the Actual member and click the Add Sibling button.
  - b. Enter the member Actual @ Bud XChg and press Enter twice.

The newly added Actual @ Bud XChg member does not consolidate into the Dollars member and must, therefore, be tagged appropriately.

4. Apply the “Exclude member from consolidation” tag to the Actual @ Bud XChg member by pressing the No Consolidation button, .

5. Define members that can store the data in local currency values:
  - a. Add the member **Local** as a sibling of **Dollars**. Select the member name **Dollars** and press the **Add Child** button.
  - b. Enter the name **Local** in the Member Edit box and press **Enter** twice.

Data only needs to be captured in local currency for **Actual** and **Budget**. These are the only children to define for **Local**.
6. Define **Local** member children to capture local data:
  - a. Select the **Local** member and click the **Add Child** button.
  - b. Enter the member names **Act** and **Bud** and press **Enter** twice.

Values for the **Act** member consolidate to **Local**, but values for the **Bud** member do not.
7. Apply the “Exclude member from consolidation” tag to the **Bud** member by pressing the **No Consolidation** button, .
8. Tag the **Local** member (which does not store data) as a **Label** by selecting it and clicking the **Label** button.
9. Tag the **Scenario** dimension (which does not contain any consolidated results) as a **Label** by selecting it and clicking the **Label** button.

Figure 43-5 shows the resulting **Scenario** dimension outline after making the modifications necessary for the TBC currency conversion application.

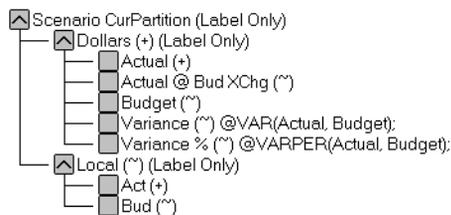


Figure 43-5: Scenario Dimension Database Outline

## Save the Main Database Outline Changes

Save the outline changes by clicking the Save button. If your database contains data values, Hyperion Essbase restructures the database to reflect changes to the outline. When you click Save, the Restructure Database dialog box shown in Figure 43-6 is displayed.

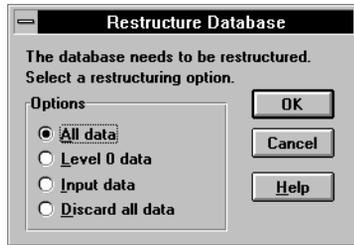


Figure 43-6: Restructure Database Dialog Box

When an existing outline is updated, you have several choices that let you restructure data values. Because no data has been loaded into the model, click OK to continue the outline update.

## Creating the Currency Database Outline

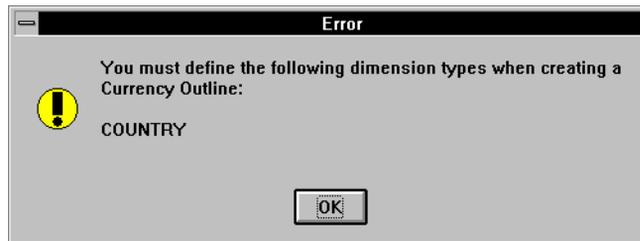
Once you have verified and saved the Main Database outline, you can generate the currency outline. The currency outline contains dimensions, members, currency names, and currency categories previously defined in your Main Database outline. It is basically structured and ready to use after being generated but may require additions to make it complete.

## Generate the Currency Database Outline

Follow these steps to generate the Currency outline:

1. Press the Generate Currency Outline button, .

If the Main outline is missing any required dimension tags (such as Time, Accounts, or Country), Hyperion Essbase displays an error box like the one shown in Figure 43-7.



*Figure 43-7: Error Dialog Box Displayed When No Country Dimension Is Defined*

2. If the Error dialog box is displayed after you attempt to generate the Currency outline, do the following:
  - a. Note the errors and click OK.
  - b. Tag the dimensions as required.

- c. Attempt to generate the Currency outline again. If all tags are correctly defined, the dialog box shown in Figure 43-8 displays.

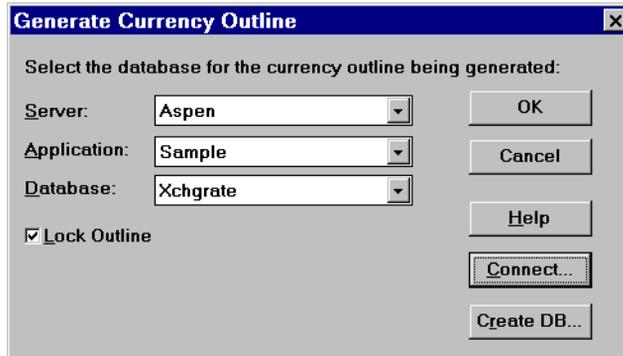


Figure 43-8: Generate Currency Outline Dialog Box

The Generate Currency Outline dialog box lets you define the database for which the outline is generated. The dialog box contains the following controls:

- The Server list box, which lets you select a Hyperion Essbase Server
- The Application list box, which lets you select an application
- The Database list box, which lets you select a database (only Currency Databases is displayed in the list)
- The Connect button, which lets you log in to a different server if necessary
- The Create DB button, which lets you create a new Currency Database

If you need to log in to a different server, click the Connect button. The dialog box shown in Figure 43-9 is displayed. If you do not need to log in to a different server, go to Step 5.

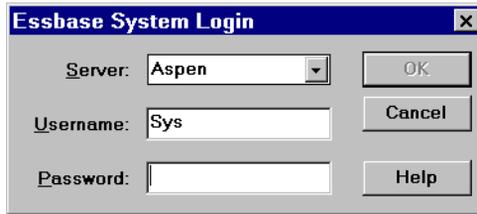


Figure 43-9: Hyperion Essbase System Login Dialog Box

3. Select the correct server, enter your user name and password, and click OK to close the dialog box and return to the Generate Currency Outline dialog box.
4. Click the Create DB button. The dialog box shown in Figure 43-10 is displayed.

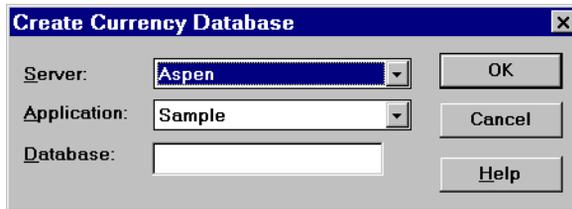


Figure 43-10: Create Currency Database Dialog Box

5. Type the database name in the Database text box. In the TBC example, we use the database name XCHGRATE.
6. Click OK to create the database.
7. When the Generate Currency Outline dialog box is displayed again, click OK to generate the outline.

The outline for the currency database XCHGRATE is displayed in the Outline Editor as shown in Figure 43-11.

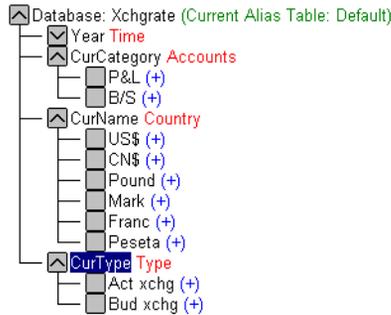


Figure 43-11: Currency Database Outline

## Review the Contents of the Currency Database Outline

After generating your Currency Database outline, review its contents and note the following items that are automatically created based on the contents of the Main Database that you defined previously:

- The Year dimension is replicated from the Main Database. This allows you to define exchange rates for each time period.
- The currency category names that were defined on some members of the Measures dimension in the Main Database become members of the CurCategory dimension.
- The currency names from the Market dimension become members of the CurName dimension.
- The CurType dimension, which contains no members, is also created. This dimension is not directly mapped to the Main Database.

## Add New Members to the CurType Dimension

After you have generated the Currency Database, you can add members to any dimension. Because the TBC Currency Database (XCHGRATE) contained different exchange rates for actual and budget, they added two new members to the CurType dimension to apply to different scenarios in the Main Database:

1. Add the Act xchg member to the CurType dimension.
2. Add the Bud xchg member to the CurType dimension.

Exchange rates from these two members are applied to different scenarios from the Main Database, as follows:

- Actual (from the Main Database) is converted with a CurType member called Act xchg.
- Actual @ Bud Xchg (from the Main Database) is converted with a CurType member called Bud xchg.
- Budget (from the Main Database) is converted with a CurType member called Bud xchg.

**Note:** For details about the currency conversion calculation process, see “Calculating Currency Conversions” on page 43-26.

## Save the Currency Database Outline Changes

When you have reviewed the newly generated Currency Database outline and have made any necessary additions, you need to save your changes.

1. Save changes made to the Currency Database outline by pressing the Save button. The dialog box shown in Figure 43-12 is displayed.

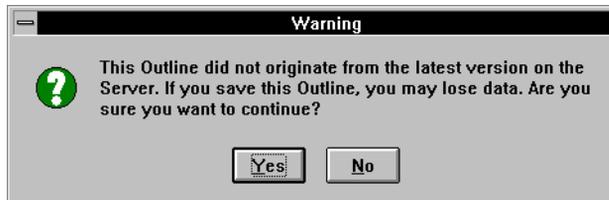


Figure 43-12: Using the Latest Version on the Server Warning

Because the newly-generated outline contains all new members, you are given a choice of continuing or canceling the operation.

2. Click Yes to continue. The Restructure Database dialog box shown in Figure 43-13 is displayed.

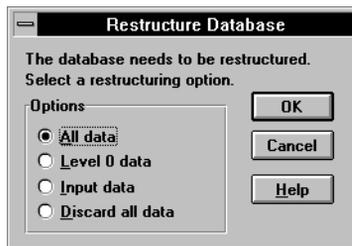


Figure 43-13: Restructure Database Dialog Box

When an existing outline is updated, you have several choices that let you restructure data values.

3. Because no data has been loaded into the model, click OK to continue the outline update.
4. Close the Outline Editor for the Main and Currency Databases.

## Linking the Main and Currency Databases

To perform a currency conversion calculation, Hyperion 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.

- To link a Currency Database to the Main Database:
  1. Select the Main Database from the application's database list.
  2. Choose Database > Settings. The General page of the Database Settings dialog box shown in Figure 43-14 is displayed.

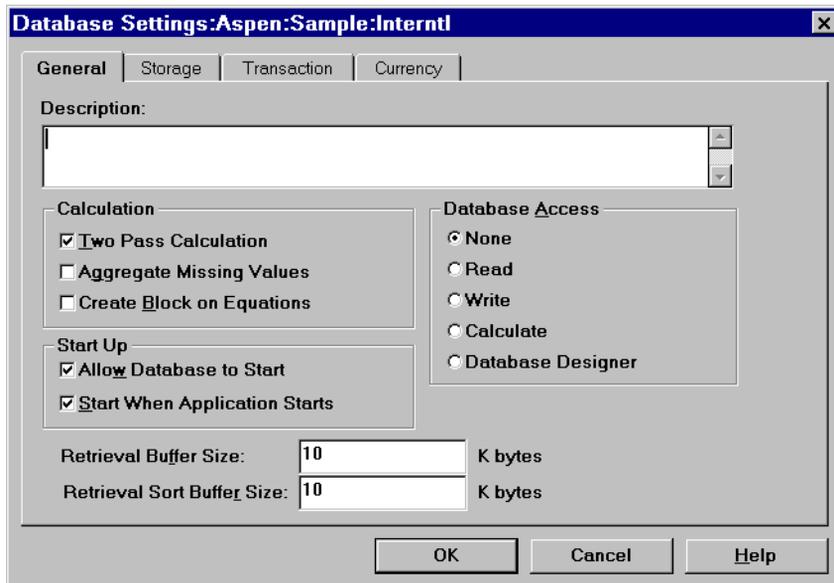


Figure 43-14: Database Settings Dialog Box General Page

3. Select the Currency page. The Currency page shown in Figure 43-15 is displayed.

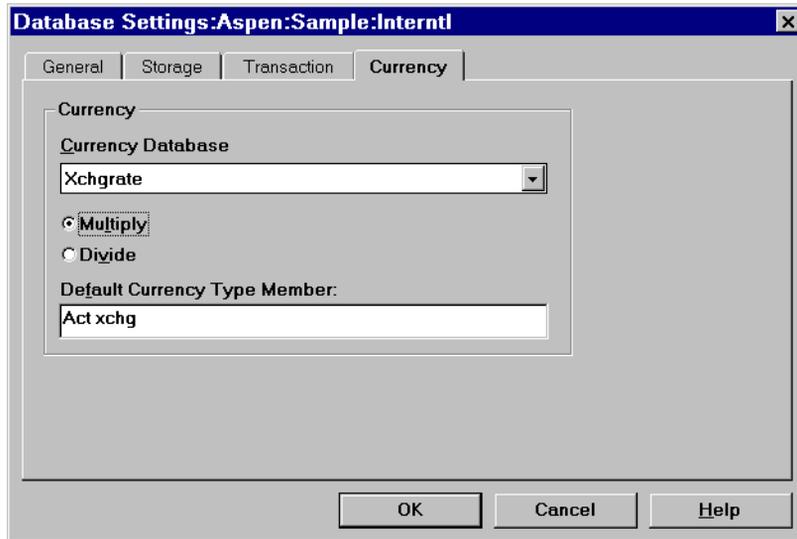


Figure 43-15: Database Settings Dialog Box Currency Page

4. Select a database from the Currency Database list box.

The Multiply and Divide options are enabled. Use these option buttons to define the conversion calculation method to be used. The Multiply option button multiplies each local data value by the exchange rate. The Divide option button divides each local data value by the exchange rate.

5. Select the Multiply option.

The Default Currency Type Member text box lets you define a currency type member to use as a default for currency conversion calculations.

6. Type **Act xchg** to assign this as the default.
7. Click OK twice to close the dialog box and save these settings.

After the databases are defined and linked, the next step is to load data into the Main Database and exchange rates into the Currency Database. After values have been entered, a currency conversion is calculated. This is accomplished by running a calc script (see “Calculating Currency Conversions” on page 43-26).

## Calculating Currency Conversions

You convert data values from a local currency to a common, converted currency using the `CCONV currExchMbr` command in a calc script. For example, you might convert data from a variety of European currencies into US\$.

**Note:** You cannot use the `CCONV` command to convert data in a transparent partition.

You convert the data values back to the original, local currencies using the `CCONV TOLOCALRATE CurType` command.

You can convert all or part of your main database using the rates defined in your Currency database. You can overwrite the local values with the converted values, or you can keep both the local and converted values in your main database, depending on your tracking and reporting needs.

## Overwriting Local Values with Converted Values

If you want to overwrite the local values, you do *not* need to create a `CURPARTITION` dimension in your main database. Use the `CCONV` command in a calc script to convert all the data in your database.

The following calc script converts the values in the database to US\$.

```
CCONV US$;  
CALC ALL;
```

If required, you can specify a currency name that contains the required exchange rate. The following calc script converts the values in the database to US\$, using the exchange rate for Jan as defined in the Currency database.

```
CCONV Jan->US$;  
CALC ALL;
```

The `CALC ALL` command is required in the examples shown, because the `CCONV` command only converts currencies. It does not consolidate or calculate members in your database.

The following calc script converts the values back to their original values in their local currencies using the "Act xchg" rate:

```
CCONV TOLOCALRATE "Act xchg";  
CALC ALL;
```

For more information on the CCONV command, see the online *Technical Reference* in the DOCS directory.

**Note:** You cannot use the FIX command unless you are using a CURPARTITION dimension and the CCTRACK setting is TRUE in the ESSBASE.CFG file.

## Keeping Local and Converted Values

43

You can keep both local and converted values in your database. In your main database you need to define the members that store the local and converted values. You do this by creating a CurPartition dimension (see “Modify the Scenario Dimension” on page 43-14). The CurPartition dimension has two partitions, one for local values and one for converted values.

You then need to create a calc script to complete the following steps:

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 CurPartition dimension, you must FIX on a member of this dimension to use the CCONV command.

3. Use the CALC command to recalculate your database.

The following example is based on the Sample INTERNTL database and corresponding Sample XCHGRATE Currency database. Figure 43-16 shows the Currency Partition from the Sample INTERNTL database.



Figure 43-16: Calculating Local and Converted Currency Conversions

The following calc 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 calc script converts the Actual and Budget values back to their original values in their local currencies:

```

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. This means that Hyperion Essbase recalculates all the converted blocks when you recalculate your database. For more information on Intelligent Calculation, see Chapter 34, “Using Intelligent Calculation to Optimize Calculation.”

## Converting Data to a Different Currency in Reports

You can calculate currency conversions in report scripts, using the `CURRENCY target currency` command to set the output currency and currency type. For the syntax and definitions of Report Writer commands, see the online *Technical Reference* in the DOCS directory.

**Note:** Hyperion Essbase cannot perform “on the fly” currency conversions across transparent databases. If you have two transparent partition databases that are calculated using different conversions, you cannot calculate 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 Qtrl
<Currency "Jan->Peseta->Act xchg"
<Ichildren Colas
    !
{CurHeading}
Illinois Sales Budget
    <Column (Year)
        <children Qtrl
    !
```

## Effects of CTRACK Parameter on Conversion Calculations

The CTRACK setting in your ESSBASE.CFG file controls whether exchange rates are tracked while Hyperion Essbase calculates currency conversions.

When CTRACK is True, Hyperion Essbase tracks exchange rates that are applied to data as conversions are calculated, allowing conversion to occur at report time through the Spreadsheet Add-in or the Hyperion Essbase Report Writer.

Setting CTRACK to False turns off the tracking system and has the following results:

- A CCONV command assumes the data to be converted is always in local currency.
- Currency report options assume that the data to be converted is always in local currency. This means that even if the data has already been converted in the database, it is reconverted at report time and can yield inaccurate results.
- The restrictions on operations such as FIX or DATACOPY are removed for conversion operations. For example, when you have a currency partition and CTRACK is set to True, you can't use the CCONV command outside of a FIX. Conversely, when you do not have a currency partition and CTRACK is set to TRUE, Hyperion Essbase does not allow you to calculate without a FIX. Setting CTRACK to False eliminates these restrictions.

For more information, refer to online *Technical Reference* in the DOCS directory.

Part  
**IX**

# Maintaining Hyperion Essbase Applications

Part IX describes how to maintain existing Hyperion Essbase OLAP Server applications, including how to use ESSCMD; run Hyperion Essbase, applications, and databases; monitor performance; work with Hyperion Essbase files in cross-platform environments; and back up and recover databases. Part IX contains the following chapters:

- Chapter 44, “Performing Interactive and Batch Operations Using ESSCMD,” describes how to use ESSCMD’s interactive and batch processing modes. It contains detailed script and batch files.
- Chapter 45, “Running Hyperion Essbase, Applications, and Databases,” describes how to use the Agent to start and stop Hyperion Essbase servers, use the server console, start and stop Hyperion Essbase databases, view a list of users, view a list of available ports, and describes how the Agent handles client/server communications.
- Chapter 46, “Using Diagnostics to Monitor Performance,” describes how to use diagnostic tools to analyze problems, solve problems, check Hyperion Essbase configuration, and analyze the Hyperion Essbase performance.
- Chapter 47, “Working with Hyperion Essbase Files and Cross-Platform Environments,” introduces you to the kinds of files that Hyperion Essbase uses, describes operations you can perform on Hyperion Essbase files, and discusses some of the issues you face when working with Hyperion Essbase files on different platforms.
- Chapter 48, “Backing Up Data and Recovering Databases,” describes when Hyperion Essbase performs commit operations, how to recover a database, how to back up a database, and other ways Hyperion Essbase protects your data.



# Performing Interactive and Batch Operations Using ESSCMD

The Hyperion Essbase OLAP Server includes ESSCMD, a command-line interface that performs operations interactively or through a batch or script file.

This chapter describes how to use ESSCMD's interactive and batch processing modes, and provides sample script and batch files. It includes the following topics:

- “ESSCMD Basics” on page 44-1
- “Quick Start to ESSCMD” on page 44-5
- “Using Interactive Mode” on page 44-6
- “Using Script and Batch Files for Batch Processing” on page 44-9

For the syntax and usage of each ESSCMD command, see the online *Technical Reference* in the DOCS directory.

## ESSCMD Basics

With ESSCMD, you can execute 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 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.

## Syntax Guidelines

In general, use the same syntax for entering ESSCMD commands as you do for other Hyperion Essbase commands. However, there are differences between ESSCMD's 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. Be sure to 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. This is especially important in order to signal the end of the parameter list if some of the 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 online *Technical Reference* in the DOCS directory, are examples of commands which you should terminate with a semicolon to prevent any confusion in processing.

**Note:** All syntax examples in this chapter use quotation marks and semicolon terminators.

## Running ESSCMD on Different Operating System Platforms

ESSCMD operates independently of any other Hyperion Essbase client interface, including the Application Manager, Spreadsheet Add-in, or custom-built application programs.

ESSCMD is available on the following platforms:

- Windows NT server
- Windows 2000 server
- HP-UX server
- AIX server
- Solaris server
- Windows 95 and Windows 98 server

## 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 object or file names with a numeric parameter, from 1 to 4, that tells Hyperion Essbase where to look for the object or file. The parameter directs ESSCMD to look for files in other applications, databases, or systems.

The following table lists each value for the numeric parameter (*numeric*), 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.

<b>numeric</b>	<b>File</b>	<b>Hyperion Essbase prompts user for:</b>
1	Local or client-based file	Windows: Files in the \ESSBASE\CLIENT\ <i>appName</i> \ <i>dbName</i> directory  UNIX: Files in the <i>essbase/client/appName/dbName</i> directory
2	Remote or server-based file	Windows: Files in the \ESSBASE\APP\ <i>appName</i> \ <i>dbName</i> directory  UNIX: Files in the <i>essbase/app/appName/dbName</i> 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 the Hyperion Essbase server. The command requires the numeric parameter to tell Hyperion 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 both 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 object is in the directory from which you started ESSCMD, you do not need to enter a path.

## Multi-User Considerations

Because ESSCMD supports multiple login instances on an Hyperion 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.

## Case-Sensitivity

Case-sensitivity varies by operating system:

- Windows platforms are not case-sensitive. You can enter ESSCMD commands and file-names in uppercase or lowercase letters, or in any combination of the two.
- UNIX is case-sensitive. You must enter file names in the correct case or UNIX does not recognize them. However, you can enter ESSCMD command names and parameters in uppercase or lowercase.

## Getting Help

To display a list of all currently available ESSCMD commands, enter `HELP ?`. To see documented descriptions and syntax for individual ESSCMD commands, see the online *Technical Reference*, located in the `ESSBASE/DOCS` directory.

## Quick Start to ESSCMD

Before you start ESSCMD, make sure that the following items are properly installed and running:

- Hyperion Essbase server
- Communications protocol (Named Pipes or TCP/IP) on the Hyperion Essbase server

For information on protocols supported by Hyperion Essbase, see the *Hyperion Essbase Installation Guide*.

## Starting and Quitting ESSCMD

The Hyperion Essbase server installation places the `ESSCMD.EXE` and `ESSCMD.HELP` files (`ESSCMD` and `esscmd.hlp` on UNIX platforms) in the `BIN` directory of your application server.

Once you start the application, a command prompt like this one displays:

```
:: : [n] ->
```

where *n* is the value of the active login instance. Each subsequent, successful login increments this value by one. When you start ESSCMD, 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. For more information, see the online *Technical Reference*, located in the `ESSBASE/DOCS` directory.

### To start ESSCMD:

Do one of the following:

- Enter `ESSCMD` at the operating system command prompt. ESSCMD runs within the operating system command line window.
- Run `ESSCMD.EXE` (`ESSCMD` on UNIX platforms), located in the `BIN` directory

### 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. This 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 44-9 for information.

For syntax conventions when working in interactive mode, see “Syntax Guidelines” on page 44-2.

## Logging Into the Server

After starting ESSCMD, you must connect to the Hyperion Essbase server so that you can enter commands. Follow these steps:

1. At the ESSCMD prompt, log in to the server with the **LOGIN** command. For more information about this command, see the online *Technical Reference*, located in the `ESSBASE/DOCS` directory.
2. Enter the Hyperion Essbase 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 as follows:

```
aspen:::userName[1]->
```

where *userName* is your login name.

You can enter any valid ESSCMD command. For a complete listing of commands, type **HELP**.

**Note:** To load an application into memory and select a database on the Hyperion Essbase server, use the **SELECT** command to select a database from an application that resides on the server. For more information, see the online *Technical Reference*, located in the `ESSBASE/DOCS` directory.

The ESSCMD prompt is displayed as follows:

```
aspen:appName:dbName:userName[1]->
```

where:

- *appName* is the name of the application.
- *dbName* is the name of the database to which you are connected.

## Entering Commands

There are two ways to enter commands in interactive mode. Choose either of the following methods to enter commands:

- Type 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, ESSCMD 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, then press Enter.

Using SELECT as the example, you would type:

```
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 Hyperion Essbase is processing a command. Pausing the system may prevent Hyperion 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.

For syntax conventions when working in batch processing mode, see “Syntax Guidelines” on page 44-2.

- 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 the Hyperion Essbase server from the operating system prompt; the file is processed by the operating system. On Windows NT, 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 order until it reaches the end of the file.

Existing script or batch files might be affected by changes to some commands. To ensure that your files work in Hyperion Essbase 6, check on your use of changed or deleted commands. See the *Hyperion Essbase Start Here* booklet for information about new and changed commands.

## 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 Hyperion 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

Enter the following command at the operating system prompt:

```
ESSCMD scriptFileName.SCR
```

Replace *scriptFileName* with the name of the script file. For example, type the following if the script file is in the current directory:

```
ESSCMD TEST.SCR
```

If the script file is in another directory, include the path. For example:

```
ESSCMD C:\WORK\SCRIPTS\TEST.SCR (an absolute path on Windows NT)
```

or

```
ESSCMD .\SCRIPTS\TEST.SCR (a relative path on Windows NT)
```

## Handling Command Errors in a Script File

ESSCMD's 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; this is called *non-zero status*.

For error checking within an ESSCMD script file, ESSCMD provides the following error-handling commands:

- **IFERROR** checks the previously executed command for a non-zero 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, `ESSCMD` 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

EXIT;
```

**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 online *Technical Reference*, located in the `ESSBASE/DOCS` directory.

## Sample Script Files

The following script files demonstrate common Hyperion Essbase batch operations. All samples are based on the Sample Basic database that comes with your Hyperion Essbase program. The scripts for these examples are available in your `\ESSBASE\APP\SAMPLE\BASIC` directory. On UNIX systems, the examples are available from `/home/hyperion/essbase/app/Sample/Basic`.

## Sample Script: Importing and Calculating Data

Suppose you need a file that executes the following actions:

- Logs in to the Hyperion 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.
- Exits ESSCMD.

The following script file does the job:

```
LOGIN "Poplar" "TomT" "Password";
SELECT "Sample" "Basic";
DISABLELOGIN;
IMPORT 2 "ACTUALS" 4 "Y" 2 "ACTUAL" "N";
CALCDEFAULT;
ENABLELOGIN;
EXIT;
```

On Windows, this script file, `SAMPLE1.SCR`, is available in the `\ESSBASE\APP\SAMPLE\BASIC` directory. On UNIX platforms, `SAMPLE1.SCR` is in the `/essbase/app/Sample/Basic` directory.

## Sample Script: Building Dimensions and Importing/Calculating Data from a SQL Source

Suppose you need a script file that executes the following actions:

- Logs in to the Hyperion 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.
- Exits ESSCMD.

The following script file does the job:

```
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;
```

On Windows, this script file, `SAMPLE2.SCR`, is available in the `\ESSBASE\APP\SAMPLE\BASIC` directory. On UNIX systems, `SAMPLE2.SCR` is in the `/essbase/app/Sample/Basic` directory.

## Sample Script: Scheduling Report Printing

Suppose you need a file that executes the following actions:

- Logs in to the Hyperion Essbase server.
- Selects an application and database.
- Assigns reports that output to files for later printing.
- Exits ESSCMD.

The following script file does the job:

```
LOGIN "Poplar" "TomT" "Password";
SELECT "Sample" "Basic";
RUNREPT 2 "REP1" "REP1.OUT";
RUNREPT 2 "REP2" "REP2.OUT";
RUNREPT 2 "REP3" "REP3.OUT";
EXIT;
```

On Windows, this script file, `SAMPLE3.SCR`, is available in the `\ESSBASE\APP\SAMPLE\BASIC` directory. On UNIX platforms, `SAMPLE3.SCR` is in the `/essbase/app/Sample/Basic` directory.

## 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.

For an example of a batch file, see “Handling Command Errors in a Script File” on page 44-11.

## 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. For information about handling errors in script files, see “Handling Command Errors in a Script File” on page 44-11.

For example, a batch file could contain three scripts: an ESSCMD batch file that loads data, a calc 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 going on. 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 an NT 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%==goto Error
ESSCMD CALC.SCR
If not %errorlevel%==goto Error
ESSCMD REPORT.SCR
If not %errorlevel%==goto Error
Echo All operations completed successfully
EXIT

:Error
Echo There was a problem running the script
```

# Running Hyperion Essbase, Applications, and Databases

Hyperion Essbase includes a central organizing subsystem called the Hyperion Essbase Server Agent that performs many important functions. When you load the main Hyperion Essbase executable file, ESSBASE, the Agent is started. The Agent is responsible for various activities, including checking security, establishing connections between the server and its clients, and accessing databases and applications.

The Agent enables you to perform many important administrative functions. This chapter describes the following Agent-related activities:

- “About the Agent” on page 45-1
- “Starting and Stopping the Hyperion Essbase Server” on page 45-2
- “Using the Server Console” on page 45-4
- “Starting and Stopping Applications” on page 45-5
- “Starting and Stopping Databases” on page 45-8
- “Viewing a List of Users and Available Ports” on page 45-10
- “Understanding Client-Server Communications” on page 45-12

## About the Agent

The Agent allows communication between the Hyperion Essbase server and its clients. As a traffic light at a busy intersection controls the flow and direction of many cars at once, the Agent controls the flow of activity during Hyperion Essbase operation. Additionally, the Agent provides you ongoing information about all actions performed by users and provides you the ability to monitor the behaviour of Hyperion Essbase.

You have access to the Agent from the server machine, which is also called the server console. Access to the Agent is available only on the server machine. After you start Hyperion Essbase, the Agent window is displayed. In the Agent window on the server console, you view the activities of the Agent and enter commands that you want the Agent to perform.

## Starting and Stopping the Hyperion Essbase Server

You can start the Hyperion Essbase server from the server console, from a terminal connected to the server, or as a background process. You can stop the server from the server console, from a terminal connected to the server, or from a batch script.

### Starting the Hyperion Essbase Server in the Foreground

To start the Hyperion Essbase server, type `ESSBASE password` at the operating system command line, where *password* is the password to access the server. This action starts the Hyperion Essbase Server Agent (see “Using the Server Console” on page 45-4).

### Starting the Hyperion Essbase Server in the Background

To start the Hyperion Essbase server in the background on platforms running Windows or UNIX, enter the following:

```
ESSBASE password -b &
```

You can start Hyperion Essbase without using the ampersand (&) at the end of the command, but if you do not use “&,” the command prompt is not returned after the server is started.

## Improving Parallel Login Processing

To speed up parallel login processing with the new command-line parameter, `quicklogin`, enter the following:

```
ESSBASE password -quicklogin -b
```

The parameter `quicklogin` improves performance by caching the security file and writing it to disk at specified time intervals.

## Stopping the Hyperion Essbase Server from the Agent

To stop the server and all running applications from the server console, enter `QUIT` or `EXIT` in the Agent window.

If you stop the Agent by closing the Agent window or by pressing `Ctrl + C`, the next time you start the database Hyperion Essbase rolls back any transactions that were in progress (see Chapter 42, “Ensuring Data Integrity,” for information about transaction processes).

45

## Stopping the Hyperion Essbase Server Remotely

Use the `SHUTDOWNSERVER` command to shut down the Hyperion Essbase server from `ESSCMD` or from a batch script. You must have supervisor privilege to use this command. For information on privilege levels, see Chapter 17, “Managing Security at Global and User Levels.”

### Using `ESSCMD` to Shut Down the Server



You can use the `SHUTDOWNSERVER` command in `ESSCMD` to shut down the server. See the online *Technical Reference* in the `DOCS` directory for information about this command. See Chapter 44, “Performing Interactive and Batch Operations Using `ESSCMD`” for information about `ESSCMD`.

## Using an ESSCMD Batch Script to Shut Down the Server

To shut down the server from an ESSCMD batch script, enter the following command in the batch script:

```
SHUTDOWNSERVER "servername" "username" "password" ;
```

In the syntax above, *servername* is the name of the server that you want to shut down; *username* is your user name; and *password* is your password.

## Using the Server Console

When Hyperion Essbase starts on the server, the main process, ESSBASE, is the Agent. The Agent can be accessed only from the server machine, which is also called the server console.

On Windows platforms, when the Agent starts, a new icon displays in the taskbar.

The Agent records its activities, such as user logins and starting applications in the Agent log file, ESSBASE.LOG (Essbase.log on UNIX systems). The log is stored in the Hyperion Essbase installation directory, for example, \ESSBASE (/essbase on UNIX systems).

From the Agent, you can use the commands described in the following table:

*Table 45-1: Agent Commands*

Command	Function
START <i>appname</i>	Starts the specified application.
STOP <i>appname</i>	Stops the specified application.
USERS	Displays a list of all users that are connected to the server. The total number of connections is displayed, as well as the number of ports available.
LOGOUTUSER <i>user</i>	Disconnects a user from the server and frees a port.  This command requires the Hyperion Essbase system password.

Table 45-1: Agent Commands (Continued)

Command	Function
PASSWORD	Changes the system password that is required to start the Hyperion Essbase server. Changing this password does not change the password of the Hyperion Essbase system supervisor.  This command requires the Hyperion Essbase system password.
VERSION	Displays the server software version number.
HELP	Lists all valid Agent commands and their respective functions.
PORTS	Displays the number of ports that are installed on the server and the number of ports that are in use.
DUMP <i>filename</i>	Dumps information from the Hyperion Essbase security system to a specified file in ASCII format. If you do not supply a path with the file name, the file is saved to the BIN directory, for example, \ESSBASE\BIN (/essbase/bin on UNIX systems).  This command requires the Hyperion Essbase system password.
QUIT and EXIT	Shuts down all open applications and quits Hyperion Essbase.

You can press Enter when the Agent window on the server console is open (Hyperion Essbase must be started in the foreground) to display a list of all available commands.

## Starting and Stopping Applications

When you start an application, Hyperion Essbase loads the application and any associated databases into memory on the server. This action makes data accessible to end users. When you stop an application, Hyperion Essbase unloads all information and databases from memory on the server.

## Starting an Application

To start an application from the Agent, type `START appname` at the command prompt on the server console.

You can also start an application by completing any of the following actions:

- Creating a new application, which automatically starts the application.
- Using the Application > Start/Stop command from Hyperion Essbase Application Manager

If the application is not already loaded, a confirmation box is displayed. Click Yes to load the application.

If the application is already loaded, the confirmation box asks if you want to stop the application. Click No to prevent Hyperion Essbase from stopping the application.

- Starting a database within an application, as discussed under “Starting and Stopping Databases” on page 45-8
- Saving an outline to the server

**Note:** Opening an outline does *not* start an application.



You can use the `LOADAPP`, `LOGIN`, or `SELECT` command in `ESSCMD` to start an application. See the online *Technical Reference* in the `DOCS` directory for information about these commands. See Chapter 44, “Performing Interactive and Batch Operations Using `ESSCMD`” for information about `ESSCMD`.

## Stopping an Application

When you stop an application, transactions may be currently running. If you issue the `STOP` command in the Agent or an `UNLOADAPP` command in `ESSCMD`, the application does not stop if a calculation or data load is in progress. Instead, Hyperion Essbase displays a message in the server console. If you stop the Agent by closing the Agent window or by pressing `Ctrl + C`, the application stops, and the next time you start the application, Hyperion Essbase rolls back any transactions that were in progress (see Chapter 42, “Ensuring Data Integrity,” for information about transaction processes).

To stop an application from the Agent, type `STOP appname` at the command prompt on the server console.

You can also stop an application by using the Application > Start/Stop command from Hyperion Essbase Application Manager:

- If the application is already loaded, a confirmation box is displayed. Click Yes to stop the application.
- If the application is not already loaded, the confirmation box asks if you want to start the application. Click No to prevent Hyperion Essbase from starting the application.



You can use the UNLOADAPP command in ESSCMD to stop an application. See the online *Technical Reference* in the DOCS directory for information about these commands. See Chapter 44, “Performing Interactive and Batch Operations Using ESSCMD” for information about ESSCMD.

**Note:** The Windows NT Task Manager does not display process IDs for individual Hyperion Essbase applications: all of the running Hyperion Essbase applications is displayed as undifferentiated `essvr.exe` processes. This prevents you from stopping a single application, in the event that the application hangs.

You can find the process ID for individual servers in the `essbase.log` file in your `ESSBASE` directory. When the server starts, a line like the following is displayed in the log file:

```
Application [Sample] started with process id [225]
```

If an application hangs, you can stop the application by using the process ID number in conjunction with a kill operating-system utility. Such a utility is provided with Windows NT Resource Kit and with various other toolkits.

On UNIX platforms, you can use the `ps` output to identify individual applications. If an application hangs, you can stop the application by using the `kill -q <pid>` command.

## Starting and Stopping Databases

Starting a database loads the database into memory on the server. Stopping a database unloads all information for the database from memory on the server.

### Starting a Database

You can start a database by completing any of the following actions:

- Creating a new database, which automatically starts the database
- Using the Database > Start/Stop command from Hyperion Essbase Application Manager

If the database is not already loaded, a confirmation box is displayed. Click Yes to load the database.

If the database is already loaded, the confirmation box asks if you want to stop the database. Click No to prevent Hyperion Essbase from stopping the database.

- Saving an outline to the server
  - Note:** Opening an outline does *not* start a database.
- Clicking the Statistics tab in the Database Information dialog box

**Note:** When you start a database from an application that is not loaded, the application is loaded along with all its related databases.



You can use the LOADDDB, LOGIN, or SELECT command in ESSCMD to start a database. See the online *Technical Reference* in the DOCS directory for information about this command. See Chapter 44, “Performing Interactive and Batch Operations Using ESSCMD” for information about ESSCMD.

### Stopping a Database

When you stop a database, transactions may be currently running. If you issue the STOP command in the Agent, or an UNLOADDB command in ESSCMD, the database does not stop if a calculation or data load is in progress. Instead, Hyperion Essbase displays a message in the Agent window. If you stop the Agent by closing the Agent window or by pressing Ctrl + C, the database stops, and the next time

you start the database, Hyperion Essbase rolls back any transactions that were in progress (see Chapter 42, “Ensuring Data Integrity,” for information about transaction processes).

To stop a database from the Agent, type `STOP appname` at the command prompt on the server console.

This action stops the application that contains the database that you want to stop.

You can also stop a database by completing any of the following actions:

- Using the Database > Start/Stop command from Hyperion Essbase Application Manager

If the database is already loaded, a confirmation box is displayed. Click Yes to stop the database.

If the database is not already loaded, the confirmation box asks if you want to start the database. Click No to prevent Hyperion Essbase from starting the database.

- Using the Application Start/Stop command from Hyperion Essbase Application Manager to stop the application that contains the database that you want to stop

If the application is already loaded, a confirmation box is displayed. Click Yes to stop the application.

If the application is not already loaded, the confirmation box asks if you want to start the application. Click No to prevent Hyperion Essbase from starting the application.



You can use the `UNLOADDB` command in `ESSCMD` to stop a database. See the online *Technical Reference* in the `DOCS` directory for information about this command. See Chapter 44, “Performing Interactive and Batch Operations Using `ESSCMD`” for information about `ESSCMD`.

## Viewing a List of Users and Available Ports

The Agent enables you to view a list of all users that are connected to the server at any given time. Additionally, you can view the total number of ports available, as well as the number of existing connections.

To view a list of all users connected to the server, type the command `USERS` at the command prompt of the server console. The server console displays the following information:

- The names of all users connected to the server
- The total number of ports available
- The total number of existing connections
- The application to which each user is connected
- The database to which each user is connected

To view the number of ports installed on the server, as well as the number of ports in use, type `PORTS` at the command prompt of the server console.



You can use the `LISTUSERS` command in `ESSCMD` to view a list of users that have been defined on the selected server. Using this command does not return information about the available ports and connections or the application or database to which users are connected. See the online *Technical Reference* in the `DOCS` directory for information about this command. See Chapter 44, “Performing Interactive and Batch Operations Using `ESSCMD`” for information about `ESSCMD`.

## Disconnecting a User From the Server

To disconnect a user from the server, type the command `LOGOUTUSER user` at the command prompt of the server console. The user’s port is now free.



You can use the `LOGOUTUSER` command in `ESSCMD` to disconnect a user from the server. See the online *Technical Reference* in the `DOCS` directory for information about this command. See Chapter 44, “Performing Interactive and Batch Operations Using `ESSCMD`” for information about `ESSCMD`.

## Changing the System Password

You can change the password that is required to start the Hyperion Essbase server from the server console of the Agent. Changing the system password does not change the connection password that is established for the Hyperion Essbase system supervisor.

To change the Hyperion Essbase system password:

1. Type **PASSWORD** at the command prompt of the server console.

Hyperion Essbase prompts you to enter the old system password, as shown in Figure 45-1.



```

essbase.exe
password
Executing command: PASSWORD
Enter old system password: *****
Enter new system password: *****
Re-type new system password: *****
[Wed Nov 12 21:11:11 1997]Local/ESSBASE0///Info(1051060)
System password updated
  
```

Figure 45-1: Changing the Hyperion Essbase System Password

2. Type the old (current) system password.
3. Type the system password that you want to use.
4. Retype the system password that you want to use.

Hyperion Essbase verifies that the system password has been updated.

For information about changing passwords for users or groups, see Chapter 17, “Managing Security at Global and User Levels.”



You can use the **SETPASSWORD** command in **ESSCMD** to change a password. See the online *Technical Reference* in the **DOCS** directory for information about this command. See Chapter 44, “Performing Interactive and Batch Operations Using ESSCMD” for information about **ESSCMD**.

## Displaying the Server Software Version Number

To see the version number of the Hyperion Essbase server software, type `VERSION` at the command prompt of the server console.



You can use the `GETVERSION` command in `ESSCMD` to display the server software version number. See the online *Technical Reference* in the `DOCS` directory for information about this command. See Chapter 44, “Performing Interactive and Batch Operations Using `ESSCMD`” for information about `ESSCMD`.

## Viewing Security System Information

To output information from the Hyperion Essbase security system into a specified file in ASCII format, type `DUMP filename` at the command prompt of the server console. Hyperion Essbase prompts you for the Hyperion Essbase system password. By default Hyperion Essbase saves the file to the `ARBORPATH\BIN` directory, where `ARBORPATH` is the Hyperion Essbase installation directory, for example, `D:\ESSBASE\BIN`.

## Shutting Down All Open Applications and Quitting Hyperion Essbase

To shut down all open applications and quit Hyperion Essbase, from the Agent window on the server console, type `QUIT` or `EXIT` at the command prompt.

## Understanding Client-Server Communications

The Hyperion Essbase client-server architecture supports enterprise analysis applications. The server runs a Server Agent (`ESSBASE`) process that acts as a traffic coordinator for all user requests to Hyperion Essbase applications. The

application server (ESSVR) fields data requests from clients. An Essnet library connects Hyperion Essbase to the network and enables the server and the clients to communicate with each other, as shown in Figure 45-2.

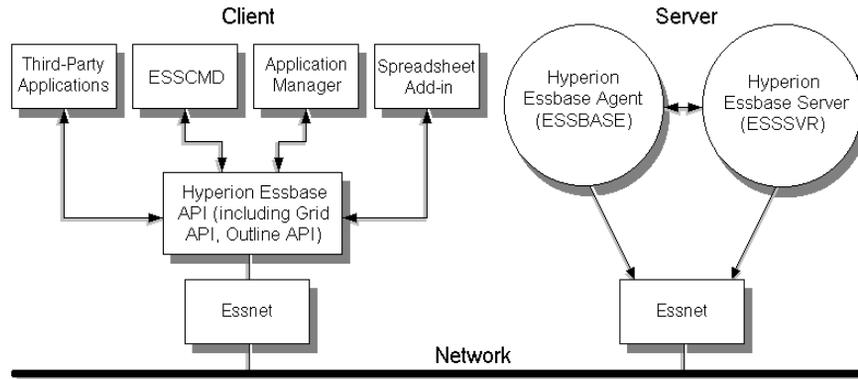


Figure 45-2: Hyperion Essbase Client-Server Communications

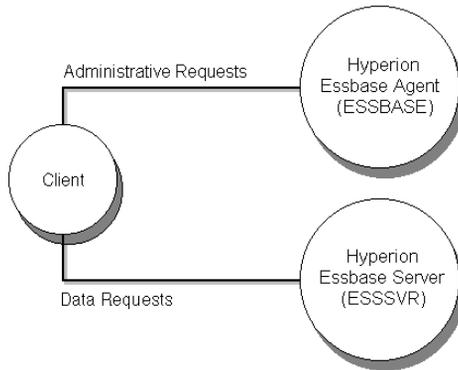
## Hyperion Essbase Client Types

Hyperion Essbase clients retrieve and analyze data from the server with Lotus 1-2-3, Microsoft Excel, or a custom application interface. There are four types of Hyperion Essbase clients:

- Hyperion Essbase Spreadsheet Add-in, which provides you with seamless data access to the server.
- Hyperion Essbase Application Manager, which you can use to design, develop, and maintain Hyperion Essbase Applications.
- ESSCMD, a command-line interface, which you can use to perform operations interactively or through a batch or script file.
- A custom application built with the Hyperion Essbase Application Programming Interface (API). The online *API Reference* in the `DOCS` directory provides a complete listing of Hyperion Essbase API functions, platforms, and supported compilers.

As shown in Figure 45-3, Hyperion Essbase handles client requests differently, depending on the type of the request. Administrative requests, such as logging in and logging out, starting and stopping applications and databases, and viewing

user security information, are handled by the Hyperion Essbase Server Agent (ESSBASE). Client requests for data, such as data loads, calculations, spreadsheet reports, and data lock and unlock, are handled by the application server (ESSSVR).



*Figure 45-3: Hyperion Essbase Client Request Handling*

## Flow of Communications Events

When a user logs in to Hyperion Essbase, the following events occur:

1. The Hyperion Essbase client logs in, using a predefined address:
  - For TCP/IP connections, the address is port number 1423.
  - For Named Pipes connections, the address is `\\pipe\essbase0`.
2. The user selects an application/database combination.
3. The Agent compares the requested application with applications currently running. If the specified application is already running, the Agent does not need to do anything. If the requested application is not already running, the Agent initiates startup, and the following events occur:
  - a. The Agent assigns a dynamic port for the application server or creates a dynamic name for a named pipe.
  - b. The application server returns the port number for the application to the Agent and to the client. Port number information is stored at run time in Hyperion Essbase API.

4. The Agent sets the application server as active with the current user and security information. When the client later sends a data request to the application server, the security information captured by the API is embedded in the request.
5. The Agent sends the client query to the application server (ESSSVR).
6. For every request that the client makes, the application server causes the following events to occur:
  - a. Connect
  - b. Send request
  - c. Get response
  - d. Send data
  - e. Receive data
  - f. Disconnect

## Multithreading

ESSBASE and ESSSVR are multithreaded applications. The number of threads is based on the number of licensed ports that are purchased, as shown in Table 45-2. The number of ports represents the number of concurrent connections that Hyperion Essbase supports. Hyperion Essbase provides one reserve port for the system administrator. The system administrator uses the reserve port to log out one or more users when all other ports are in use.

*Table 45-2: Licensed Ports and Multithreading*

Number of Licensed Ports	Number of Threads
1–5 ports	5
6–10 ports	10
11 or more ports	20

You can set the number of threads for the Agent or the server in the `ESSBASE.CFG` file. For complete details on setting the number of threads for the Agent and the server, refer to the `AGENTTHREADS` and `SERVERTHREADS` settings in the online *Technical Reference* in the `DOCS` directory.

## Application Monitoring

Each application that is loaded is an open task or process in the operating system. On Windows platforms, the application is displayed in a Hyperion Essbase server window. On UNIX platforms, the application server is a child process of ESSBASE. When the application starts, ESSBASE starts the ESSSVR process.

The application server records its activities (such as writing data to an application log file, *appname*.LOG) in the *appname* directory below the Hyperion Essbase installation directory, for example, the `\ESSBASE\APP\appname` directory. You can open and view this text file when you need to troubleshoot problems or view application activity.

On Windows platforms you can also view application activities as they occur in the Hyperion Essbase server window. On UNIX, you can view application activities with the `tail -f logfile` command.

On Windows platforms, when the Agent starts an application, a new icon is displayed in the taskbar. You can double-click the icon to view the server window.

Chapter  
**46**

# Using Diagnostics to Monitor Performance

---

This chapter describes the Hyperion Essbase OLAP Server diagnostic tools available to you and tells where to find log files that can help you analyze and solve problems. Use this section not only when problems occur but also when you want to check or analyze Hyperion Essbase configuration or performance.

This chapter contains the following sections:

- “Overview of Diagnostic Tools” on page 46-2
- “Quick Reference to Diagnostic Tools” on page 46-18
- “Exception Error Handling” on page 46-26
- “Application Event Log File” on page 46-28
- “Server Event Log File” on page 46-31
- “Creation of a Directory for Trace Files” on page 46-33
- “Outline Change Log File” on page 46-35
- “Object Locks” on page 46-40
- “Application Monitoring” on page 46-41
- “Server Error Message Categories” on page 46-41

## Overview of Diagnostic Tools

Hyperion Essbase application, server, and database information dialog boxes in Hyperion Essbase Application Manager are useful when you are preparing to migrate data, add users, analyze performance problems, or perform other administrative tasks. These dialog boxes are available from the Server, Application, and Database menus in Hyperion Essbase Application Manager. For details about accessing specific information in dialog boxes and a review of the pages that dialog boxes contain, see Table 46-1.

**Note:** For information about a specific field in any dialog box, display the dialog box and press the Help button.

## Viewing Application Information

The Application Information dialog box as shown in Figure 46-1, provides information about an application and its databases.

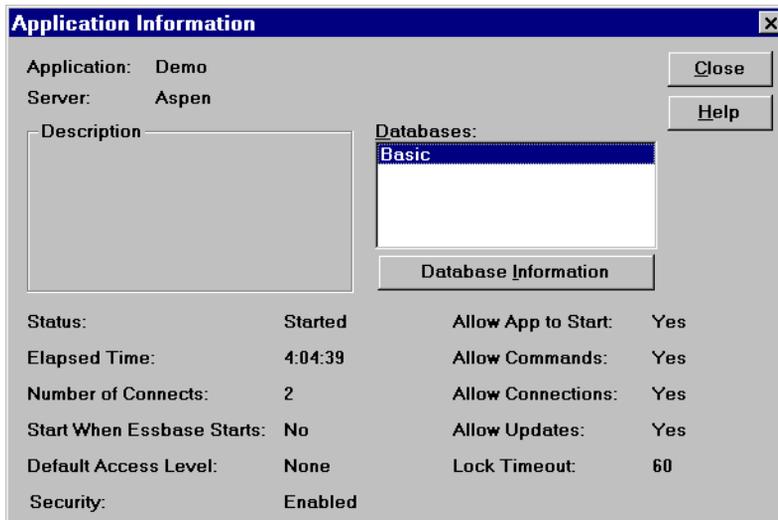


Figure 46-1: Application Information Dialog Box

Use the Application Information dialog box to verify which databases are running in the application and to check access, security, and start-up information.

You can also display the Database Information dialog box (see “Viewing Database Information” on page 46-9) from the Application Information dialog box by selecting a database in the list box and clicking the Database Information button.

The settings and other information in the Application Information dialog box are for information only. If you want to customize your application, you can change the settings in the Application Settings dialog box, as described in Chapter 17, “Managing Security at Global and User Levels.”

Click the Help button for information on each item.



You can use the GETAPPSTATE command in ESSCMD to perform this task. See the online *Technical Reference* in the DOCS directory for information about this command. See Chapter 44, “Performing Interactive and Batch Operations Using ESSCMD” for information about ESSCMD.

## Viewing Server Information

The Server Information dialog box provides information, grouped by tabs, on server license, configuration, operating system, disk drives, and applications.

Hyperion Essbase displays information on a snapshot basis; to see the latest information, click the Refresh button for a new snapshot. The Refresh button refreshes every page in the dialog box, not just the current page.

The settings and other information in the Server Information dialog box are for information only and reflect activity on the server. You cannot use the dialog box to change settings; the purpose of the dialog box is to help you identify activities or operations that may be affecting performance.

## Viewing License Information

Use the License Info page to see information about the Hyperion Essbase installation; for example, you can verify whether Hyperion Essbase is correctly installed and determine which options are installed.

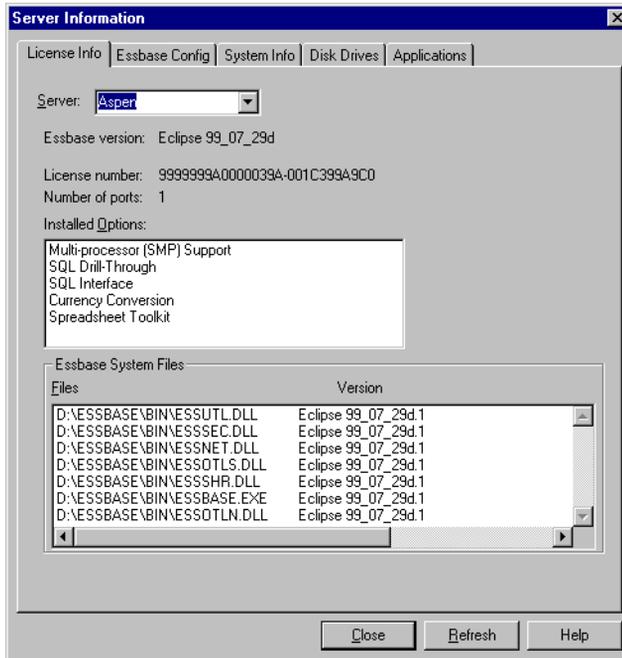


Figure 46-2: License Info Page of the Server Information Dialog Box

Click the Help button for information on each item.

## Viewing Configuration Information

Use the Essbase Config page to see information about Hyperion Essbase status and configuration. The page reflects the `ESSBASE.CFG` server file settings, if you have created an `ESSBASE.CFG` file; if you have not, the Global `ESSBASE.CFG` Settings list box is blank. See the online *Technical Reference* in the `DOCS` directory for information on how to create an `ESSBASE.CFG` file.

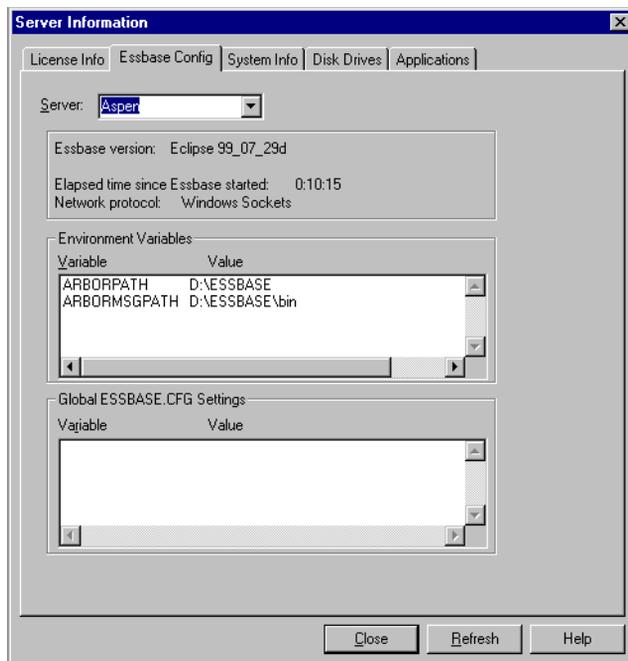


Figure 46-3: Hyperion Essbase Config Page of the Server Information Dialog Box

The Environment Variables list box displays operating system environment variables, as defined during installation. You can use this list box to verify path information.

Click the Help button for information on each item.

## Viewing System Information

Use the System Info page to review information about the operating system and about resource usage. You use this page to determine whether machine or operating system constraints are impacting Hyperion Essbase performance.

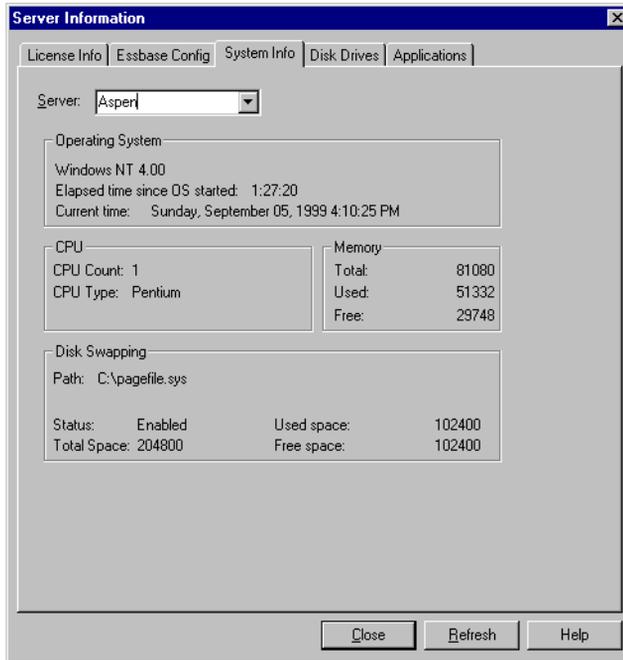


Figure 46-4: System Info Page of the Server Information Dialog Box

For example, suppose Hyperion Essbase is running too slowly. Check the Memory and Disk Swapping groups to see how much free space is available. Assume that the Free space value in Disk Swapping is very low.

Take the following action, depending your platform:

- On Windows platforms, increase the maximum virtual memory or add more volumes for disk swapping.
- On UNIX platforms, increase swap space or add another swapping device.

Click the Help button for information on each item.

## Viewing Disk Drive Information

The Disk Drives page contains information about disk drive types, disk drive usage, and file system types. Use this information to determine whether lack of disk space or incompatibility of file system types is impacting Hyperion Essbase performance.

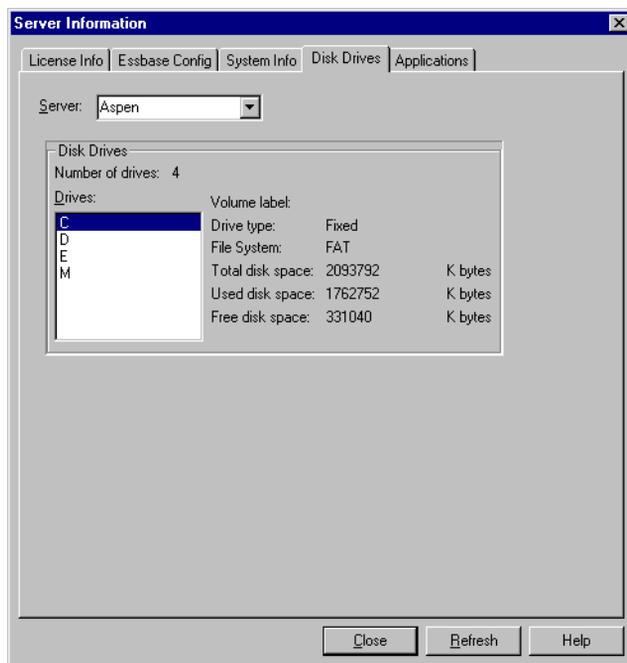


Figure 46-5: Disk Drives Page of the Server Information Dialog Box

For example, if information on the System Info page indicates that swap space is low, you can use the Disk Drives page to see which drives have space available.

You can also use the Disk Drives page to see which drives have space available when you want to allocate space on another drive. See “Specifying Disk Volumes” on page 41-20 for information about disk spanning using the Disk Volumes setting.

Click the Help button for information on each item.

## Viewing Application Status Information

Use the Applications page to see which applications and databases are loaded.

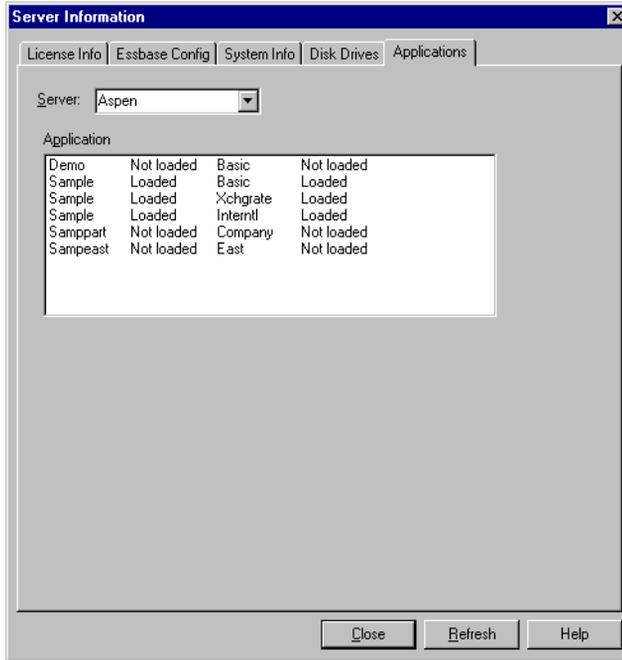


Figure 46-6: Applications Page of the Server Information Dialog Box

Hyperion Essbase lists only those applications that you are authorized to use.

Click the Help button for information on each item.



You can use the GETAPPINFO command in ESSCMD to perform this task. See the online *Technical Reference* in the DOCS directory for information about this command. See Chapter 44, “Performing Interactive and Batch Operations Using ESSCMD” for information about ESSCMD.

## Viewing Database Information

The Database Information dialog box provides information, grouped by tabs, such as database storage, database statistics, and lock contention. To see the Database Information dialog box, select a database and then select Database > Information from Hyperion Application Manager.

The server name, application name, and database name are displayed on every page. Hyperion Essbase displays information on a snapshot basis; to see the latest information, click the Refresh button for a new snapshot. The Refresh button refreshes every page in the dialog box, not just the current page.

The settings and other information in the Database Information dialog box are for information only and reflect activity on the server. You cannot use the dialog box to change settings; the purpose of the dialog box is to help you identify activities or operations that may be affecting performance.

Click the Help button for information on each item.



You can use the GETDBINFO command in ESSCMD to perform this task. See the online *Technical Reference* in the DOCS directory for information about this command. See Chapter 44, “Performing Interactive and Batch Operations Using ESSCMD” for information about ESSCMD.

For information on changing database settings, see Chapter 41, “Specifying Hyperion Essbase Kernel Settings.”



You can use the SETDBSTATEITEM or SETDBSTATE command in ESSCMD to perform this task. See the online *Technical Reference* in the DOCS directory for information about these commands. See Chapter 44, “Performing Interactive and Batch Operations Using ESSCMD” for information about ESSCMD.

## Viewing General Database Information

Use the General page to check database status and start-up information.

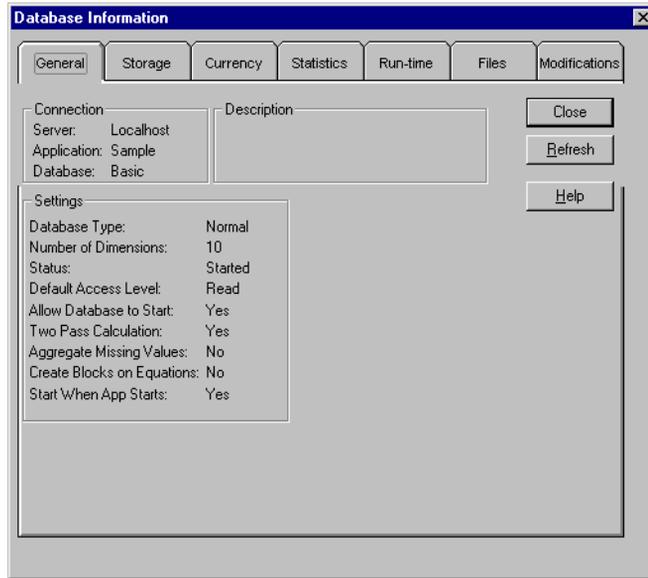


Figure 46-7: General Page of the Database Information Dialog Box

Click the Help button for information on each item.

## Viewing Database Storage Information

Use the Storage page to check data storage information, including cache sizes and the data compression setting.

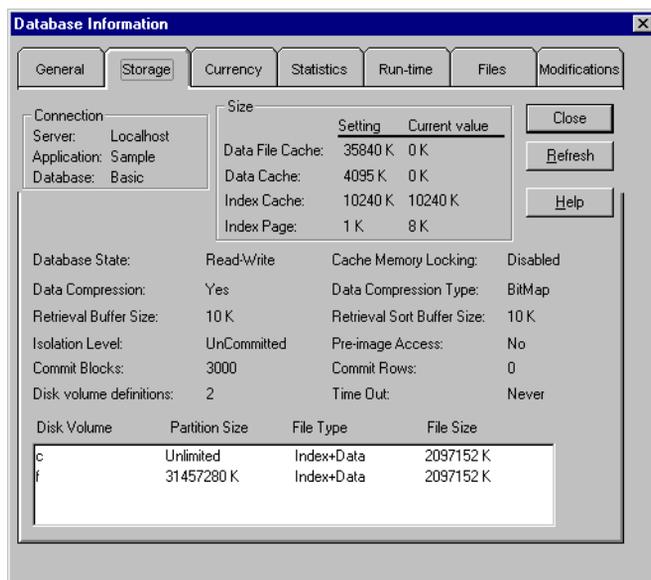


Figure 46-8: Storage Page of the Database Information Dialog Box

For more information, see Chapter 41, “Specifying Hyperion Essbase Kernel Settings.” Click the Help button for information on each item.



You can use the GETDBSTATE command in ESSCMD to perform this task. Using this command also provides currency database information that is on a separate (Currency) page of the Database Information dialog box that you access from the Application Manager desktop. See the online *Technical Reference* in the DOCS directory for information about this command. See Chapter 44, “Performing Interactive and Batch Operations Using ESSCMD” for information about ESSCMD.

## Viewing Currency Database Information

Use the Currency page to examine the configuration of the currency database associated with the current database, if the current database is linked to a currency database.

**Note:** Hyperion Essbase Currency Conversion is an optional module, licensed separately. It may not be installed on your server.



Figure 46-9: Currency Page of the Database Information Dialog Box

Click the Help button for information on each item.



You can use the GETCRDBINFO command in ESSCMD to perform this task. See the online *Technical Reference* in the DOCS directory for information about this command. See Chapter 44, “Performing Interactive and Batch Operations Using ESSCMD” for information about ESSCMD.

## Viewing Database Statistics

Use the Statistics page for dimension information and to see characteristics of data blocks in your database.

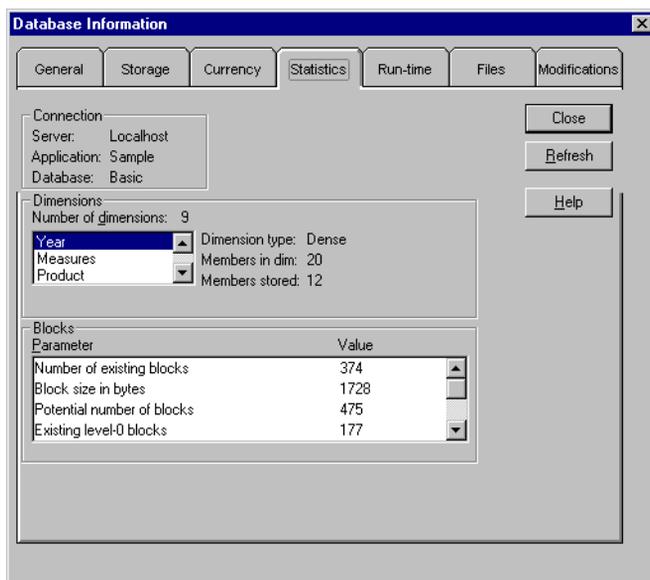


Figure 46-10: Statistics Page of the Database Information Dialog Box

To check on block density and usage, in the Blocks list box, scroll down to see the “Block density” and “Percentage of maximum blocks existing” fields.

“Block density” is the average percentage of cells containing data in a datablock; “Percentage of maximum blocks existing” is a comparison between existing blocks and potential blocks, expressed as a percentage.

If you have a low block density and a high percentage of maximum blocks, you may want to increase the number of dense dimensions. See Chapter 4, “Basic Architectural Elements,” for background information on dimension types.

To check block size, look at the “Block size in bytes” field. This field reports the size of the expanded block, whether or not you have data compression enabled. Ideally, this value should be between eight kilobytes and 64 kilobytes. To alter block size, you need to change your dense or sparse configuration.

The Compression Ratio indicates the ratio of the compressed block size to the expanded block size. For more information about data compression, see Chapter 40, “Introducing the Hyperion Essbase Kernel” and Chapter 41, “Specifying Hyperion Essbase Kernel Settings.”

If the database is not started when you choose the Statistics tab, Hyperion Essbase prompts you to start the database.

Click the Help button for information on each item.



You can use the GETDBSTATS command in ESSCMD to perform this task. See the online *Technical Reference* in the DOCS directory for information about this command. See Chapter 44, “Performing Interactive and Batch Operations Using ESSCMD” for information about ESSCMD.

## Viewing Run-Time Information

The Run-time page displays run-time information such as lock contention, insufficient cache size (buffer shortage), hit ratio on index searches (percentage of searches that did not involve retrieving from disk), and the number of read and write operations.

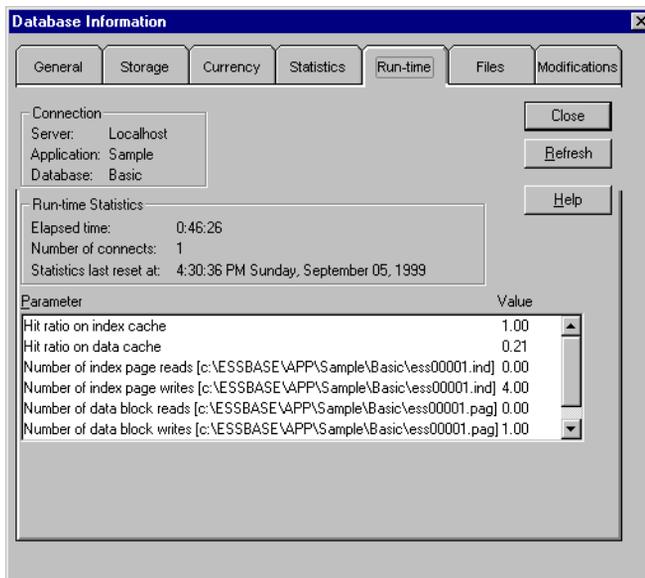


Figure 46-11: Run-Time Page of the Database Information Dialog Box

For example, suppose you want to know if your index cache size is large enough. One indicator to check is the “Hit ratio on index cache” field (you need to scroll down in the Parameters list box to see it). This value indicates the success rate of Hyperion Essbase in locating index data in memory without having to retrieve it from the disk. Ideally, this value should be near 100 percent. If the number of hits is very low, the index cache (buffer) is too small. For information on setting the index cache size, see “Using Hyperion Essbase Application Manager for Database Settings” on page 41-5.

If the database is not started when you choose the Run-time tab, Hyperion Essbase prompts you to start the database.

Click the Help button for information on each item.



You can use the GETDBINFO command in ESSCMD to display some of this information. See the online *Technical Reference* in the DOCS directory for information about this command. See Chapter 44, “Performing Interactive and Batch Operations Using ESSCMD” for information about ESSCMD.

## Viewing Database File Information

The Files page lists the index and data (.PAG) files associated with the current application and database.

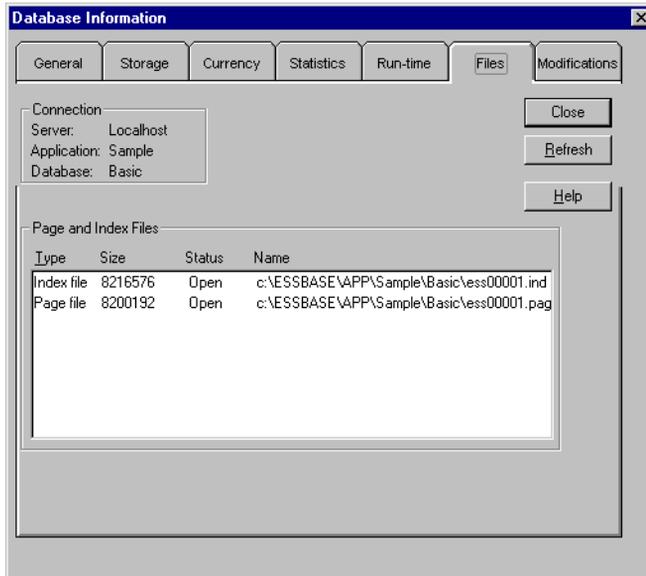


Figure 46-12: Files Page of the Database Information Dialog Box

Use this page to see where index and data (.PAG) files are stored and to see whether the files span multiple volumes.

**Note:** If the database in the Database field is started, the index files and data files should be open. If they are closed, first click the Refresh button. If the files are still closed, a problem may have occurred when the database started. Check your application and server log files (see “Application Event Log File” on page 46-28 and “Server Event Log File” on page 46-31) to see whether you can determine the problem. If you cannot, call the technical support department of your software provider. You can also check the Applications page of the Server Information dialog box (see “Viewing Application Status Information” on page 46-8) to see if the database is started.

Click the Help button for information on each item.

## Viewing a Record of Database Modifications

Use the Modifications page to see information about the last successful data load, about calculation, and about outline update operations for the current application.

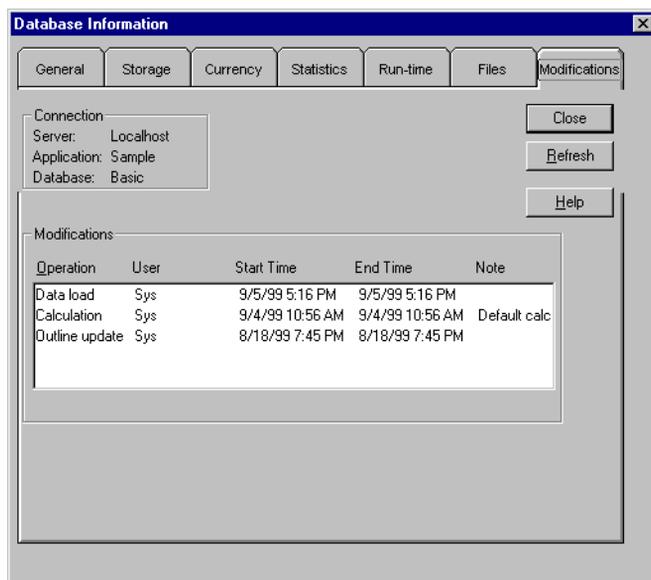


Figure 46-13: Modifications Page of the Database Information Dialog Box

**Note:** Start Time and End Time include the necessary preparation that Hyperion Essbase does prior to the operation, such as locking the data. To see the duration of the actual operation itself, see the “Elapsed time” entry in the application log. “Application Event Log File” on page 46-28 describes this log and where to find it.

Click the Help button for information on each item.

## Quick Reference to Diagnostic Tools

Table 46-1 provides detailed procedures for accessing the correct dialog boxes to perform common server, database, and application maintenance tasks. Use this table to look up the proper menu commands and dialog box pages to choose when you want to view specific information that is displayed in the Server Information or Database Information dialog boxes.

*Table 46-1: Quick Reference to Diagnostic Tools*

<b>To do this...</b>	<b>Open this dialog box and page</b>
To see a list of the servers you are currently connected to	Select Server > Information to display the Server Information dialog box. Select the License Info page to view the Server list box.
To view the version of Hyperion Essbase that is currently running on the server	Select Server > Information to display the Server Information dialog box. Select the License Info page to view the Essbase Version item.
To check the license number of your copy of Hyperion Essbase	Select Server > Information to display the Server Information dialog box. Select the License Info page to view the License number item.
To see the total number of ports that can connect to the server	Select Server > Information to display the Server Information dialog box. Select the License Info page to view the Number of ports item.
To see a list of all features in place when Hyperion Essbase was installed	Select Server > Information to display the Server Information dialog box. Select the License Info page to view the features list under Installed Options.
To see a list of the system files currently in the server memory, as well as their locations and version numbers	Select Server > Information to display the Server Information dialog box. Select the License Info page to view the Hyperion Essbase System Files list box.
To see elapsed time since Hyperion Essbase started	Select Server > Information to display the Server Information dialog box. Select the Hyperion Essbase Config page to view the Elapsed time since Essbase started item.
To check the type of network protocol the server is using	Select Server > Information to display the Server Information dialog box. Select the Essbase Config page to see the Network protocol.

Table 46-1: Quick Reference to Diagnostic Tools (Continued)

To do this...	Open this dialog box and page
To view environment variables, including source path information	Select Server > Information to display the Server Information dialog box. Select the Essbase Config page to see the Environment Variables list box.
To view the version of the operating system on which Hyperion Essbase is running	Select Server > Information to display the Server Information dialog box. Select the System Info page to view the Operating System item.
To see the time that has elapsed since the operating system was started or rebooted (in hours:minutes:seconds format)	Select Server > Information to display the Server Information dialog box. Select the System Info page to view the Elapsed time since OS started item.
To see the time and date the Server Information was retrieved, according to the server	Select Server > Information to display the Server Information dialog box. Select the System Info page to view the Current time item.
To see the number and type of CPUs in the system on which the server is operating	Select Server > Information to display the Server Information dialog box. Select the System Info page to view the CPU group.
To see the amount of available memory on the operating system	Select Server > Information to display the Server Information dialog box. Select the System Info page to view the Memory group.
To see the disk swap path	Select Server > Information to display the Server Information dialog box. Select the System Info page to view the Path item in the Disk Swapping group.
To check if disk swapping is enabled	Select Server > Information to display the Server Information dialog box. Select the System Info page to view the Status item in the Disk Swapping group.
To see the total swap space	Select Server > Information to display the Server Information dialog box. Select the System Info page to view Total Space in the Disk Swapping group.
To see the swap file size	Select Server > Information to display the Server Information dialog box. Select the System Info page to view the Used space in the Disk Swapping group.
To see available swap space	Select Server > Information to display the Server Information dialog box. Select the System Info page to view Free space in the Disk Swapping group.

Table 46-1: Quick Reference to Diagnostic Tools (Continued)

<b>To do this...</b>	<b>Open this dialog box and page</b>
To view the drive type (Fixed, Removable, RAM, Remote, or Unknown)	Select Server > Information to display the Server Information dialog box. Select the Disk Drives page and select the drive in the Drives list box. The Drive type item displays the drive type.
To view the file system used by a drive (FAT, HPFS, NTFS, etc.)	Select Server > Information to display the Server Information dialog box. Select the Disk Drives page and select the drive in the Drives list box. The File System item displays the type of file system used by the drive.
To view the total amount of space on a drive	Select Server > Information to display the Server Information dialog box. Select the Disk Drives page and select the drive in the Drives list box. The Total disk space item displays the total disk space.
To view the amount of space being used on a drive	Select Server > Information to display the Server Information dialog box. Select the Disk Drives page and select the drive in the Drives list box. The Used disk space item displays the amount of space currently being used by the selected drive.
To view the amount of space available on a drive	Select Server > Information to display the Server Information dialog box. Select the Disk Drives page and select the drive in the Drives list box. The Free disk space item displays the amount of space currently available on the selected drive.
To see which applications and databases are loaded	Select Server > Information to display the Server Information dialog box. Select the Applications page. The Application list box displays the following information: <ul style="list-style-type: none"> <li>• A list of the applications that you are authorized to use</li> <li>• The status of the listed applications (Loaded, Not Loaded, Loading, and Unloading)</li> <li>• A list of available databases</li> <li>• The status of each database (Loaded, Not Loaded, Loading, and Unloading)</li> </ul>
To view your database connections	Select Database > Information to display the Database Information dialog box. Select the General page. The server, application, and database currently loaded are displayed in the Connection group.

Table 46-1: Quick Reference to Diagnostic Tools (Continued)

To do this...	Open this dialog box and page
To view your database type	Select Database > Information to display the Database Information dialog box. Select the General page. Database Type is displayed in the Settings group.
To view the number of dimensions in a database	Select Database > Information to display the Database Information dialog box. Select the General page. The Number of Dimensions is displayed in the Settings group.
To view the status of a database	Select Database > Information to display the Database Information dialog box. Select the General page. Status (for example, Stopped) is displayed in the Settings group.
To view the default access level to a database	Select Database > Information to display the Database Information dialog box. Select the General page. The Default Access Level is displayed in the Settings group.
To see whether users may start the database	Select Database > Information to display the Database Information dialog box. Select the General page. The Allow Database to Start status is displayed in the Settings group.
To see if your database aggregates missing values	Select Database > Information to display the Database Information dialog box. Select the General page. The Aggregate Missing Values status is displayed in the Settings group.
To see if your database creates blocks on equations	Select Database > Information to display the Database Information dialog box. Select the General page. The Create Blocks on Equations status is displayed in the Settings group.
To see if a database is set to start automatically when an application is started	Select Database > Information to display the Database Information dialog box. Select the General page. The Start When App Starts status is displayed in the Settings group.
To view the data file cache size of a database (in kilobytes)	Select Database > Information to display the Database Information dialog box. Select the Storage page. The Setting and Current Value of the data file cache are displayed in the Size group.
To view the data cache size of a database (in kilobytes)	Select Database > Information to display the Database Information dialog box. Select the Storage page. The Setting and Current value of the data cache are displayed in the Size group.

*Table 46-1: Quick Reference to Diagnostic Tools (Continued)*

<b>To do this...</b>	<b>Open this dialog box and page</b>
To view the index cache size of a database (in kilobytes)	Select Database > Information to display the Database Information dialog box. Select the Storage page. The Setting and Current value of the index cache are displayed in the Size group.
To see if the cache memory locking option is enabled in a database	Select Database > Information to display the Database Information dialog box. Select the Storage page. The Cache Memory Locking status is displayed.
To view the index page size of a database (in kilobytes)	Select Database > Information to display the Database Information dialog box. Select the Storage page. The Setting and Current value of the index pages are displayed in the Size group.
To see if the compression option is enabled in a database	Select Database > Information to display the Database Information dialog box. Select the Storage page. The Data Compression status is displayed.
To see if you are connected to a currency database	Select Database > Information to display the Database Information dialog box. Select the Currency page. See Currency Database in the Currency group
To see if the currency conversion option is on in a currency database	Select Database > Information to display the Database Information dialog box. Select the Currency page. See Conversion option in the Currency group.
To see if a dimension is set to a specific country in a currency database	Select Database > Information to display the Database Information dialog box. Select the Currency page. See Country Dimension in the Currency group.
To view the time dimension setting in a currency database	Select Database > Information to display the Database Information dialog box. Select the Currency page. See Time Dimension in the Currency group.
To view the category dimension setting in a currency database	Select Database > Information to display the Database Information dialog box. Select the Currency page. See Category Dimension in the Currency group.
To view the dimension setting of the currency partition	Select Database > Information to display the Database Information dialog box. Select the Currency page. See Currency Partition Dimension in the Currency group.

Table 46-1: Quick Reference to Diagnostic Tools (Continued)

To do this...	Open this dialog box and page
To see the default type of currency that a member is set to in a currency database	Select Database > Information to display the Database Information dialog box. Select the Currency page. See Default Currency Type Mbr in the Currency group.
To view the number of dimensions in a database	Select Database > Information to display the Database Information dialog box. Select the Statistics page. See Number of dimensions in the Dimensions group.
To view a list of the dimensions in a database	Select Database > Information to display the Database Information dialog box. Select the Statistics page. All of the dimensions in the database are listed in the list box in the Dimensions group.
To see if a dimension is designated as sparse or dense	Select Database > Information to display the Database Information dialog box. Select the Statistics page. Select the specific dimension from the list box in the Dimensions group. The Dimension type (either Sparse or Dense) is displayed.
To see the number of members in a dimension	Select Database > Information to display the Database Information dialog box. Select the Statistics page. Select the specific dimension from the list box in the Dimensions group. The Members in dim item is displayed.
To see the number of stored members in a dimension	Select Database > Information to display the Database Information dialog box. Select the Statistics page. Select the specific dimension from the list box in the Dimensions group. The Members stored item is displayed.
To view the number of existing blocks in a database	Select Database > Information to display the Database Information dialog box. Select the Statistics page. In the Blocks group, select Number of existing blocks from the list box beneath the Parameters heading.
To view the block size in bytes	Select Database > Information to display the Database Information dialog box. Select the Statistics page. In the Blocks group, select Block size in bytes from the list box beneath the Parameters heading.  <b>Note:</b> Block size varies widely from block to block.

Table 46-1: Quick Reference to Diagnostic Tools (Continued)

To do this...	Open this dialog box and page
To view the potential number of blocks in a database	Select Database > Information to display the Database Information dialog box. Select the Statistics page. In the Blocks group, select Potential number of blocks from the list box beneath the Parameters heading.
To view the number of existing level-0 blocks in a database	Select Database > Information to display the Database Information dialog box. Select the Statistics page. In the Blocks group, select Existing level-0 blocks from the list box beneath the Parameters heading.
To view the number of existing upper-level blocks in a database	Select Database > Information to display the Database Information dialog box. Select the Statistics page. In the Blocks group, select Existing upper-level blocks from the list box beneath the Parameters heading.
To view block density in a database	Select Database > Information to display the Database Information dialog box. Select the Statistics page. In the Blocks group, select Block density from the list box beneath the Parameters heading. <b>Note:</b> Density varies widely from block to block.
To view a comparison between the number of existing blocks and the maximum number of possible blocks in a database	Select Database > Information to display the Database Information dialog box. Select the Statistics page. In the Blocks group, select Percentage of maximum blocks existing from the list box beneath the Parameters heading.
To view the compression ratio of blocks in a database	Select Database > Information to display the Database Information dialog box. Select the Statistics page. In the Blocks group, select Compression Ratio from the list box beneath the Parameters heading.
To view the run-time that has elapsed since the database was started, according to the server (in hours:minutes: seconds format).	Select Database > Information to display the Database Information dialog box. Select the Run-time page. See Elapsed time in the Run-time Statistics group.
To view the number of users connected to the database	Select Database > Information to display the Database Information dialog box. Select the Run-time page. See the Number of connects in the Run-time Statistics group.

Table 46-1: Quick Reference to Diagnostic Tools (Continued)

To do this...	Open this dialog box and page
To view the hit ratio on the data file, data, and index caches (the success rate of Hyperion Essbase in locating data in the appropriate cache, without having to retrieve the data from disk) and the number of index page and data block read and writes (the number of times Hyperion Essbase read index page or data block information from disk or wrote index page or data block information to disk).	Select Database > Information to display the Database Information dialog box. Select the Run-time page. Select the appropriate option from the Parameters list box.
To view the size and status of a data (page) file or an index file	Select Database > Information to display the Database Information dialog box. Select the Files page. In the Page and Index Files group, the file type (page or index) is displayed under the Type heading; the size of the file is displayed under Size; the status of the file (open or closed) is displayed under Status; and the name of the file is displayed under Name.
To view details about modifications to a database	Select Database > Information to display the Database Information dialog box. Select the Modifications page. The Operation (type of modification), the User, the Start Time and the End Time are all displayed in the list box in the Modifications group.

## Exception Error Handling

The Hyperion Essbase Exception Handler gathers information that can help the technical support department of your software provider to pinpoint where and why an error has occurred. Hyperion Essbase maintains two exception error logs, one for the application and one for the server. For additional information, see the following:

- “Application Event Log File” on page 46-28
- “Server Event Log File” on page 46-31

When an exception occurs, Hyperion Essbase displays an error message on the server console. The message states “Exception occurred,” followed by information about the name and location of the appropriate exception error log. The Process Type field in the log header indicates whether the exception occurred in the application or on the server.

## Error Log Information

The exception error log provides the following information. The information in the application log is the same as in the server log except where noted:

- Exception information
- Machine registers
- Stack and trace dumps
- Application-wide configuration
- Operating system resources
- System-wide configuration
- `ESSBASE.CFG` values
- License information (server log only)
- Client request activity
- Application information (application log only)
  - File information
  - Database information
  - Database statistics

Address of the exception pinpoints where the crash occurred.

## Error Log Names and Locations

The Hyperion Essbase Exception Handler writes information into the exception error log on the local disk in an ASCII file as follows:

- If the server crashes, the report is written in the directory specified by `ARBORPATH`; for example, `D:\ESSBASE\log00001.XCP`.
- If the application crashes and the application name is unknown, the report is written into the `APP` directory in the directory specified by `ARBORPATH`; for example, `D:\ESSBASE\APP\log00001.XCP`.
- If the application crashes and the application name is known but the database name is unknown, the report is written to the appropriate application directory; for example, `D:\ESSBASE\APP\SAMPLE\log00001.XCP`.
- If the application crashes and both the application and database names are known, the report is written to the appropriate database directory; for example, `D:\ESSBASE\APP\SAMPLE\BASIC\log00001.XCP`.
- If an internal Hyperion Essbase diagnostic utility generates a trace (`.TRC`) file, the Hyperion Essbase Exception Handler stores the file in a dump directory that you must define (see “Creation of a Directory for Trace Files” on page 46-33).

The exception error log filename is normally `log00001.XCP`. The `ExceptionLogOverwrite` parameter in the `ESSBASE.CFG` file (see the online *Technical Reference* in the `DOCS` directory) controls whether Hyperion Essbase overwrites the log file as follows:

- When `ExceptionLogOverwrite` is `TRUE` (the default), Hyperion Essbase overwrites the previous error log report with the new one.
- When `ExceptionLogOverwrite` is `FALSE`, Hyperion Essbase creates a new file instead of overwriting the previous one.

Subsequent reports are numbered sequentially; for example, if `log00001.XCP` exists, the next report has the filename `log00002.XCP`, the next has `log00003.XCP`, and so on.

## Application Event Log File

Hyperion Essbase records all application-related activity, including calculations and database restructuring, in a log file on the server. The log file resides in the application directory; for example, \ESSBASE\APP\*application\_name*. It is called *application\_name*.LOG, where *application\_name* is the name of the current application.

### Viewing the Application Log File

Hyperion Essbase Application Manager provides a facility to view the log file.

- To view the log file:
  1. If you are not connected to the appropriate server, select Server > Connect.
  2. From the Application Desktop window, select the application whose log file you want to review. Applications are listed in the Applications list box.
  3. Select Application > View Event Log.

The View Log File dialog box is displayed.

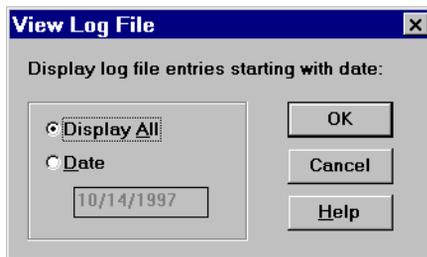


Figure 46-14: View (Application) Log File Dialog Box

4. To view the log from a specific date to the present, click the Date option and enter the starting date for log file entries. You can view the entire log file or view only those log entries beyond a specific date.

5. Click OK to view the log.

The Log Viewer is displayed, as shown in Figure 46-15. It allows you to scroll through all recorded activities.

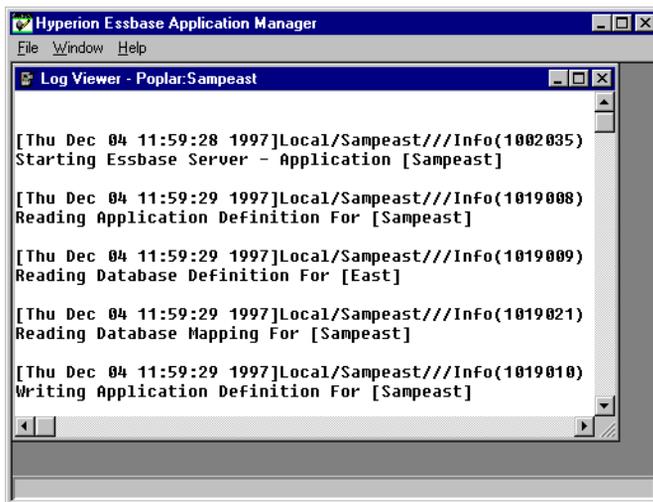


Figure 46-15: Sample Application Log

**Note:** Any activities that take place while you review the log are not reflected in the Log Viewer; the Log Viewer only presents a snapshot of the log file *at the time of your request*.

6. To close the viewer, select File > Close.

## Deleting the Application Log File

The log file uses disk space on the server. As additional activities are recorded, the log file grows. Occasionally, you may need to delete a log file before the log file grows too large.

To have Hyperion Essbase automatically clear out the log file whenever a message is written to it, enter `CLEARLOGFILE TRUE` in the `ESSBASE.CFG` server file. When `CLEARLOGFILE` is `FALSE` (the default setting), Hyperion Essbase appends new messages without overwriting existing messages. For information about the `ESSBASE.CFG` file and the syntax to use for this setting, see the online *Technical Reference* in the `DOCS` directory.

- To delete the log file:
  1. If you are not connected to the appropriate server, selectc Server > Connect.
  2. From the Application Desktop window, select the application whose log file you want to delete. Applications are listed in the Applications list box.
  3. Select Application > Delete Event Log.
  4. When Hyperion Essbase prompts you to confirm the deletion, click Yes to delete the log.

You may want to back up the log file before deleting it. You can store the log file on any disk, diskette, or tape that has sufficient available storage space. You must back up the log file from the Server console.

---

**CAUTION:** CLEARLOGFILE affects all applications on your server.

---

The View Event Log and the Delete Event Log menu commands are available only if you have at least application designer privileges for the application.



You can use the DELETEDLOG command in ESSCMD to perform this task. See the online *Technical Reference* in the DOCS directory for information about this command. See Chapter 44, “Performing Interactive and Batch Operations Using ESSCMD” for information about ESSCMD.

**Note:** The DELETEDLOG command deletes only the log file for the application that you specify.

## Considering Performance

Because the application log file grows as activities are recorded, you may want to control which activities are logged in the file.

For example, you can use the TIMINGMESSAGES parameter in the ESSBASE.CFG file to control whether Hyperion Essbase logs the duration of each spreadsheet and report query in the application log file. If this parameter is disabled, Hyperion Essbase does not log query duration times in the log file. In addition to controlling the size of the application log file, setting TIMINGMESSAGES to FALSE prevents Hyperion Essbase from having to

communicate with the operating system to obtain query start and finish times. As a result, query execution time may improve in environments with many concurrent users. For information about other `.CFG` parameters that may affect performance or log file size, see the online *Technical Reference* in the DOCS directory.

## Server Event Log File

All server-related activity is recorded in a log file on the server. The log file resides in the `ARBORPATH` directory and is called `ESSBASE.LOG`. This log file records information pertaining to all applications in the server.

### Viewing the Server Log File

Hyperion Essbase Application Manager provides a facility to view the server event log file.

- To view the log file:
  1. If you are not connected to the appropriate server, select `Server > Connect`.
  2. Select `Server > View Event Log`.

The View Log File dialog box is displayed:

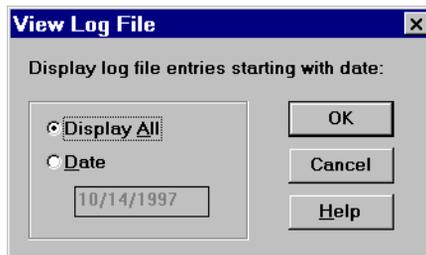


Figure 46-16: View (Server) Log File Dialog Box

3. To view the log from a specific date to the present, click the `Date` option and enter the starting date for log file entries. You can view the entire log file or view only those log entries beyond a specific date.

4. Click OK to view the log. The Log Viewer, as shown in Figure 46-17, is displayed. It allows you to scroll through all recorded activities.

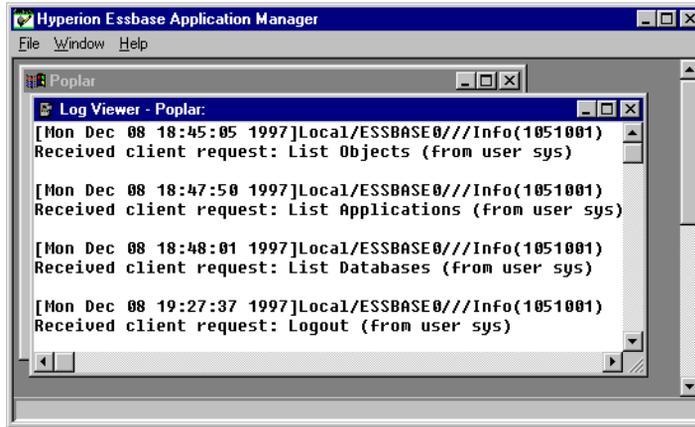


Figure 46-17: Sample Server Log

**Note:** Any activities that take place while you review the log are not reflected in the Log Viewer; the Log Viewer presents a snapshot of the log file *at the time of your request*.

Only users with supervisor authority can view the Server Log file.

5. To close the viewer, select File > Close.

## Deleting the Server Log File

The log file uses disk space on the server. As additional activities are recorded, the log file grows. Occasionally, you may need to delete a log file before it grows too large.

To have Hyperion Essbase automatically clear out the log file whenever a message is written to it, enter `CLEARLOGFILE TRUE` in the `ESSBASE.CFG` server file. When `CLEARLOGFILE` is `FALSE` (the default setting), Hyperion Essbase appends new messages without overwriting existing messages. For information about the `ESSBASE.CFG` file and the syntax to use for this setting, see the online *Technical Reference* in the `DOCS` directory.

To delete the log file:

1. If you are not connected to the appropriate server, select **Server > Connect**.
2. Select **Server > Delete Event Log**.
3. When Hyperion Essbase prompts you to confirm the deletion, click **Yes** to delete the log.

**Note:** You may want to back up the log file before deleting it. You can store the log file on any disk, diskette, or tape that has sufficient available storage space. You must back up the log file from the Server console.

Only users with supervisor authority can view the Server Log file.



You can use the `DELETELOG` command in `ESSCMD` to perform this task. See the online *Technical Reference* in the `DOCS` directory for information about this command. See Chapter 44, “Performing Interactive and Batch Operations Using `ESSCMD`” for information about `ESSCMD`.

## Creation of a Directory for Trace Files

On rare occasions, an internal Hyperion Essbase diagnostic utility may generate a trace (`.TRC`) file. When this file is generated, Hyperion Essbase creates an entry in the application event log file. The entry contains the trace file name and location and indicates an error number that identifies the exception. To capture trace files, you need to identify a directory for Hyperion Essbase to store the trace files in.

You have two choices:

- Use the `ARBORDUMPPATH` environment variable to define a location. Use this option if you have limited space on your main Hyperion Essbase server machine.
- Create a new `\ARBORPATH\DUMP` directory on your Hyperion Essbase server machine. Use this option if you want to store trace files on your main Hyperion Essbase server machine.

**Note:** Both methods are server-wide settings and need to be defined only once for each Hyperion Essbase server installation. If you define `ARBORDUMPPATH` and also create a `DUMP` directory, Hyperion Essbase uses the `ARBORDUMPPATH` directory to store the trace files.

## Defining an ARBORDUMPPATH Environment Variable

To define the ARBORDUMPPATH environment variable, decide where you want to store the trace files, such as `D:\TRACE`. Then use your operating system to define the variable as follows:

On Windows platforms, `SET ARBORDUMPPATH=D:\TRACE`

On UNIX platforms, `setenv ARBORDUMPPATH "/trace"`

Be sure that the Hyperion Essbase server has appropriate access to write files into the directory.

## Defining a DUMP Directory

If you do not create an ARBORDUMPPATH environment variable, you need to create an `\ARBORPATH\DUMP` directory on your main Hyperion Essbase server machine.

- To create a new DUMP directory on Windows platforms:
  1. Change directory to your Hyperion Essbase `%ARBORPATH%` directory.  
For example: `CD C:\ESSBASE`
  2. Create a directory called DUMP.  
For example: `MD DUMP`
  
- To create a new dump directory on UNIX platforms:
  1. Change directory to your Hyperion Essbase `$ARBORPATH` directory.  
For example: `cd /essbase`
  2. Create a directory called dump.  
For example: `mkdir essbase/dump`

## Trace File Notification

When an internal Hyperion Essbase diagnostic utility generates a trace (.TRC) file, the application event log file indicates that the file was created and identifies its location (see “Error Log Names and Locations” on page 46-27).

If you have an ARBORDUMPPATH environment variable or an \ARBORPATH\DUMP directory, Hyperion Essbase creates a trace file that is named with a long string of numbers. The numbers are separated by underscores and end with .TRC.

**Note:** If your operating system does not support long file names, the .TRC file name might be truncated.

If you have no ARBORDUMPPATH or \ARBORPATH\DUMP directory, no trace files are created. You can still benefit by reporting the error number from the application log file to the technical support department of your software provider.

## Outline Change Log File

You can set Hyperion Essbase to provide an outline change log that saves detailed modification information to an ASCII text file. You can review the outline change log at any time to see a history of all the changes that have been made to an outline since the time it was originally created. This information can be useful to you in the event you ever need to roll back an outline to a previous version. To activate the outline change log, add the following to your ESSBASE.CFG file:

```
OUTLINECHANGELOG TRUE
```

Each Hyperion Essbase database contains a separate outline change log file. The file and the database occupy the same location. Hyperion Essbase names the log file with the name of the corresponding database followed by the extension .OLG (*database\_name*.OLG). The file is stored in the database directory on the server.

The data format of the outline change log file is as follows:

- Date and time of outline modification
- Name of the user who made the change
- Type of change the user made
- Details describing the type of change made

## Outline Change Detail That is Displayed in the Log

When a user makes and saves changes to an outline, Hyperion Essbase logs modification information into the outline change log as an individual record. Each change record begins and ends with the identifying information shown in Figure 46-18:

```
[Begin Outline Change for app0/Dbl, Thu Sep 19 11:29:54, 1999,
By User1]

Added new dimension "Overseas"
dimension type - sparse
dimension tag - (none)
Left sibling - None
Status - Store Data
Level Number - 0
Generation Number - 1

[End Outline Change for app0/Dbl, Thu Sep 18 19:00:25, 1999, by User1]
[Begin Outline Change for app0/Dbl, Fri Sep 19 11:29:54, 1999, By User1]

Number of member changes for the dimension "Year" : 2

Added new member "day1" to "Jan" :
Left sibling - None
Unary calc Symbol - Add
Level Number - 0
Generation Number - 4

Added new member "day2" to "Jan" :
Left sibling - "day1"
Status - Store Data
Unary calc Symbol - Add
Level Number - 0
Generation Number - 4

[End Outline Change for app0/Dbl, Fri Sep 19 11:29:56, 1999, By User1]
```

*Figure 46-18: Sample Outline Change Log*

The nature of the detailed change information that the outline change log provides depends on the type of change the user makes to the outline, as detailed in Table 46-2. See “Impact of Common Outline Changes” on page 40-21 for information on the restructuring impact of each type of outline change.

*Table 46-2: Outline Change Log Contents*

<b>Type of Change</b>	<b>Information Included</b>
Add a dimension	<ul style="list-style-type: none"> <li>• Name of dimension</li> <li>• Type of dimension (dense or sparse)</li> <li>• Dimension tag (if any)</li> <li>• Name of left sibling<sup>1</sup> of dimension</li> <li>• Level number of dimension</li> <li>• Generation number of dimension</li> </ul>
Delete a dimension	<ul style="list-style-type: none"> <li>• Name of dimension</li> </ul>
Update a dimension	<ul style="list-style-type: none"> <li>• Name of dimension</li> <li>• Dimension tag</li> <li>• Type of dimension (dense or sparse)</li> <li>• Level name changes (if applicable)</li> <li>• Generation name changes (if applicable)</li> </ul>
Rename a dimension	<ul style="list-style-type: none"> <li>• Old name of dimension</li> <li>• New name of dimension</li> </ul>
Move a dimension to a new position	<ul style="list-style-type: none"> <li>• Name of dimension</li> <li>• Old location, including left sibling<sup>1</sup> of dimension</li> <li>• New location, including left sibling<sup>1</sup> of dimension</li> </ul>

Table 46-2: Outline Change Log Contents (Continued)

Type of Change	Information Included
Add a member to a dimension	<ul style="list-style-type: none"> <li>• Name of new member or members</li> <li>• Unary calc symbol for member</li> <li>• Level number of member</li> <li>• Generation number of member</li> <li>• Status of member (Store, Share)</li> <li>• Member alias (if applicable)</li> <li>• Account type of member</li> <li>• User-defined attributes of member (if applicable)</li> <li>• Calc formula for member</li> </ul>
Update a member of a dimension	<ul style="list-style-type: none"> <li>• Name of member updated</li> <li>• Member properties that were updated</li> </ul>
Rename a member of a dimension	<ul style="list-style-type: none"> <li>• Old name of member</li> <li>• New name of member</li> </ul>
Move a member of a dimension to a new position	<ul style="list-style-type: none"> <li>• Name of member moved</li> <li>• New location</li> <li>• Names of parent and left sibling<sup>1</sup> in new location</li> </ul>

<sup>1</sup>: 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 dimension or member as None.

## Outline Change Log On and Off Function

To activate the outline change log function, you need to make the following entry in the ESSBASE .CFG file:

```
OUTLINECHANGELOG TRUE
```

After you have turned the function on, Hyperion Essbase updates the outline change log automatically each time a user makes changes to a Hyperion Essbase database outline.

**Note:** During a restructure, Hyperion 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. “Restructuring Considerations” on page 40-20 discusses other conditions that affect restructure performance.

If you want to disable changes from being made to the outline change log file or to remove the log from the `ESSBASE.CFG` file, change the setting in the `ESSBASE.CFG` file to the following:

```
OUTLINECHANGELOG FALSE
```

## Setting for the Outline Change Log File Size

The default file size assigned for the outline change log is 64,000 bytes. When the file reaches this size, Hyperion Essbase copies the contents of the file to a backup file. The name of the backup file is the same as the outline change log file, but with an `.OLB` extension (`database_name.OLB`). The outline change log is cleared, and any additional log entries are written to the outline change log file. Each time the outline change log file reaches its maximum file size, Hyperion Essbase replaces the backup file with a backup of the current outline change log. This cycle repeats as Hyperion Essbase continues to add log entries to the outline change log.

If you want to set the outline change log file size larger than the default of 64,000 bytes, you need to make the following entry in the `ESSBASE.CFG` file:

```
OUTLINECHANGELOGFILESIZE n
```

In the syntax above, *n* is a number interpreted in bytes. For example, entering 20000 sets the file size to 20,000 bytes; entering 9216 sets the file size to 9 kilobytes. The minimum file size is 8,092 bytes; the maximum is 2 megabytes. The default, minimum, and maximum file sizes for the backup file are the same as the defaults specified for the outline change log file. For example, if you use the `OUTLINECHANGELOGFILESIZE` setting to change the maximum size of the outline change log file to 2 megabytes, you automatically set the `.OLB` backup file to the same maximum size.

## Object Locks

Hyperion Essbase uses a check-out facility for server-based objects to ensure that no more than one user attempts to modify an object at one time. By default, whenever a user opens a server-based object, Hyperion Essbase locks the object. If a second user attempts to open the same object, a message is displayed. The message indicates that the object is already locked and notes who has locked it.

## Overriding a File Lock

You can override the lock error by unchecking Lock file in the Application Desktop window before attempting to open an object. This action essentially opens the file in read-only mode and you cannot save any changes you make.

## Unlocking Objects

There are two ways to unlock a previously locked object:

- Save the locked file to the server.
- Select File > Unlock. This action unlocks a file that you previously locked but did not save to the server.

**Note:** Users can unlock only those objects that they originally locked. A supervisor can unlock files that are locked by other users.

File locking is not the same as block locking. For information about how Hyperion Essbase locks data blocks, see Chapter 42, “Ensuring Data Integrity.”



You can use the UNLOCKOBJECT command in ESSCMD to perform this task. See the online *Technical Reference* in the DOCS directory for information about this command. See Chapter 44, “Performing Interactive and Batch Operations Using ESSCMD” for information about ESSCMD.

## Application Monitoring

Each application that is loaded is an open task or process in the operating system. On Windows platforms, the application is displayed in a Hyperion Essbase server window. On UNIX platforms, the application server is often a background process. When the application starts, `ESSBASE.EXE` (ESSBASE on UNIX systems) starts the ESSVR process.

The application server records its activities, such as writing data to an application log file, `appname.LOG`, in the `\ESSBASE\APP\appname` directory. You can open and view this text file when you need to troubleshoot problems or view application activity. On Windows platforms, you can also view application activities as they occur in the Hyperion Essbase server window. On UNIX, you can route all messages to a file with the `tail -f logfile` command, where `logfile` is the name of a file that you specify.

On Windows platforms, when the Agent starts an application, a new icon is displayed in the Taskbar. You can double-click the icon to view the server window.

## Server Error Message Categories

Table 46-3 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 look up the Hyperion Essbase component to which the error is related.

Table 46-3: Server Error Message Categories

Error Message Number Range	Hyperion Essbase Component That Generated Error
1001000-1001999	Report Writer Extractor
1002000-1002999	General server
1003000-1003999	Hyperion Essbase kernel
1004000-1004999	Not currently used
1005000-1005999	Backup, export, or validate
1006000-1006999	Data cache
1007000-1007999	Member structure
1008000-1008999	System calls, portable layer, ASD, or Agent

*Table 46-3: Server Error Message Categories (Continued)*

<b>Error Message Number Range</b>	<b>Hyperion Essbase Component That Generated Error</b>
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
1018000-1018999	Message database
1019000-1019999	Database objects
1020000-102999	Spreadsheet Extractor
1021000-1021999	SQL Interface
1022000-1022999	Security
1023000-1023999	Partitioning
1024000-1024999	Query Extractor
1030000-1030999	Application Programming Interface (API)
1040000-1040999	Network
1041000-1041999	Network—Named Pipes
1042000-1042999	Network—TCP
1050000-1059999	Agent
1060000-1069999	Outline API
1070000-1079999	Index cache
1080000-1089999	Transaction manager
1090000-1099999	Rules file processing

*Table 46-3: Server Error Message Categories (Continued)*

<b>Error Message Number Range</b>	<b>Hyperion Essbase Component That Generated Error</b>
1100000-1109999	Grid API
1110000-1119999	Hyperion Web Gateway (HWG)
1140000-1149999	Linked Reporting Objects
1150000-1169999	Outline operations
1170000-1179999	Attributes
1180000-1189999	Showcase
1190000-1199999	Hyperion Integration Server (HIS)
1200000-1120999	Calculator framework



# Working with Hyperion Essbase Files and Cross-Platform Environments

This chapter introduces the file types that Hyperion Essbase uses, describes operations that you can perform on Hyperion Essbase files, and discusses some of the issues that you face when working with Hyperion Essbase files on different platforms.

This chapter contains the following sections:

- “About Hyperion Essbase File Types” on page 47-1
- “Operating on Applications, Databases, and Related Objects” on page 47-5
- “Porting Applications Across Platforms” on page 47-15

## About Hyperion Essbase File Types

The following table lists the file types that Hyperion Essbase uses to store applications and databases:

*Table 47-1: Hyperion Essbase File Types for Applications and Databases*

<b>File Extension</b>	<b>Description</b>
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

*Table 47-1: Hyperion Essbase File Types for Applications and Databases (Continued)*

<b>File Extension</b>	<b>Description</b>
chg	Outline synchronization change file
csc	Hyperion Essbase calc script
db	Database file, defining the name, location, and other database settings
dbb	Backup of database file
dbf	dBASE data file
ddb	Partitioning definition file
ddm	Temporary partitioning file
ddn	Temporary partitioning file
esm	Hyperion Essbase kernel file, containing control information that is used for database recovery
esr	Temporary database root file
esn	Temporary Hyperion Essbase kernel file
ind	Hyperion Essbase index file
inn	Temporary Hyperion Essbase index file
log	Server or application log file
lro	Linked reporting object file that is linked to a data cell
lst	Cascade table of contents or list of files to back up
mdx	dBASE multiple index file
ocl	Database change log
ocn	Incremental restructuring file
oco	Incremental restructuring file
olb	Backup of outline change log
olg	Outline change log
otl	Hyperion Essbase outline file
otm	Temporary Hyperion Essbase outline file

*Table 47-1: Hyperion Essbase File Types for Applications and Databases (Continued)*

<b>File Extension</b>	<b>Description</b>
otn	Temporary Hyperion Essbase outline file
oto	Incremental restructuring file
pag	Hyperion Essbase database data (page) file
pan	Temporary Hyperion Essbase database data (page) file
rep	Hyperion Essbase report script
rul	Hyperion Essbase data load rules file
scr	Hyperion Essbase ESSCMD script
sel	Saved member select file
str	Exported Hyperion Essbase outline file
tct	Hyperion Essbase database transaction control file
tcu	Temporary database transaction control file
trc	Trace file, created on rare occasions by an internal Hyperion Essbase diagnostic utility. Trace files can be created only if a dump file exists in the Hyperion Essbase directory.
txt	ASCII text file, such as a data file to load or a text document to link as a linked reporting object
wk4	Lotus 1-2-3 spreadsheet file
xcp	Exception error log file
xls	Microsoft Excel spreadsheet file

The following table lists the types of Hyperion Essbase files that are stored in the `\ESSBASE\BIN` directory:

*Table 47-2: Hyperion Essbase File Types in the \ESSBASE\BIN Directory*

<b>File Extension</b>	<b>Description</b>
12a	Hyperion Essbase Spreadsheet Add-in for Lotus 1-2-3
bak	Backup of security file
bnd	Microsoft ODBC file for SQL server installation using a DB2 database
cfg	Hyperion Essbase configuration file
cnt	Online help contents file
cp1	Microsoft ODBC driver for Windows platforms
dll	Microsoft Windows Dynamic Link Library
exe	Executable file
hlp	Online help file
ico	Icon bitmap image
lck	Hyperion Essbase lock file
lic	License information file for ODBC
mdb	Message database file
sec	Security file
sl	HP-UX shared library file
so	Solaris shared library file
xll	Hyperion Essbase Spreadsheet Add-in for Microsoft Excel

The following table lists the types of Hyperion Essbase files that are stored in the \ESSBASE\API sub-directories:

*Table 47-3: Hyperion Essbase File Types in the \ESSBASE\API Directory*

<b>File Extension</b>	<b>Description</b>
a	UNIX static library file
bas	Microsoft Visual Basic program source file, containing header definitions for the Hyperion Essbase API
h	C or C++ header file, containing header definitions for the Hyperion Essbase API
lib	C or C++ program library
lwp	LAN Workplace network library
np	Named Pipes network library
tcp	TCP/IP network library
w95	Windows 95 Windows Socket network library

## Operating on Applications, Databases, and Related Objects

You can use Hyperion Essbase Application Manager to copy, rename, delete, and lock each of the following:

- Applications
- Databases
- Related objects, such as outlines, calc scripts, report scripts, rules files, and data sources

Hyperion Essbase stores each application as a collection of files in the Hyperion Essbase application directory (for example, \ESSBASE\APP\*appname*) on the server. Hyperion Essbase stores each database as a collection of files on the database directory of the server (for example, \ESSBASE\APP\*appname*\*dbname*).

Files that are related to applications and databases are called *objects*. The common types of objects are outlines, calc scripts, report scripts, rules files, and data sources. You typically store objects on the server in the *appname* or *dbname* directory or on the client machine under the `\ESSBASE\CLIENT\appname` or *dbname* directory.

In addition to the standard object types, Hyperion Essbase also supports the following objects:

- Linked reporting objects, which are files or cell notes that are associated with specific data cells in a database. For more information, read Chapter 12, “Linking Objects to Hyperion Essbase Data.”
- Visual Basic programming objects (.OCX) for creating OLAP-aware modules. This feature is available with Hyperion Objects, licensed separately.

## Copying an Application

When you copy an application, Hyperion Essbase copies all files associated with the application to the destination application. Before copying, make sure that you have enough disk space to contain a full copy of the application, databases, and related files.

- To copy an application in Hyperion Essbase Application Manager:
  1. Connect to the server on which the application resides.
  2. From the Application Desktop window, select the application.
  3. Select Application > Copy. The following dialog box is displayed:

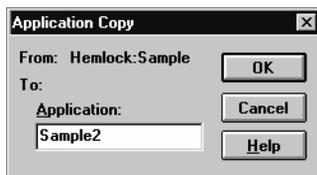


Figure 47-1: Application Copy Dialog Box

4. In the Application text box, enter the name of the copy of the application.

5. Click OK. Hyperion Essbase creates a new application and a new database directory structure on the server under the `\ESSBASE\APP` directory.



You can use the `COPYAPP` command in `ESSCMD` to perform this task. See the online *Technical Reference* in the `DOCS` directory for information about this command. See Chapter 44, “Performing Interactive and Batch Operations Using `ESSCMD`” for information about `ESSCMD`.

## Renaming an Application

When you rename an application, the application and the application directory (`ESSBASE\APP\appname`) are renamed. All objects within the application (for example databases or calc scripts) with the same name as the application are not renamed.

- To rename an application in Hyperion Essbase Application Manager:
  1. Connect to the server on which the application resides.
  2. From the Application Desktop window, select the application.
  3. Select `Application > Rename`. The following dialog box is displayed:

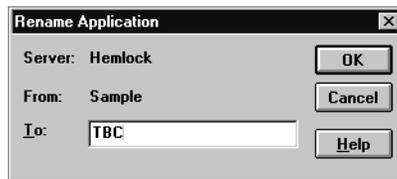


Figure 47-2: Rename Application Dialog Box

4. In the `To` text box, enter the new name for the application. The name can be up to 8 alphanumeric characters.

5. Click OK. Hyperion Essbase renames the application and directory name under the \ESSBASE\APP directory.



You can use the RENAMEAPP command in ESSCMD to perform this task. See the online *Technical Reference* in the DOCS directory for information about this command. See Chapter 44, “Performing Interactive and Batch Operations Using ESSCMD” for information about ESSCMD.

## Deleting an Application

When you delete an application, all objects within the application are also deleted. The \ESSBASE\APP\appname directory and all files located in the directory are deleted.

- To delete an application in Hyperion Essbase Application Manager:
  1. Connect to the server on which the application resides.
  2. From the Application Desktop window, select the application.
  3. Select Application > Delete. The following dialog box is displayed:

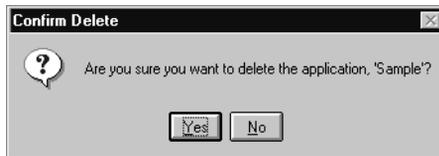


Figure 47-3: Confirm Delete Dialog Box for Applications

4. Click Yes or No.



You can use the DELETEAPP command in ESSCMD to perform this task. See the online *Technical Reference* in the DOCS directory for information about this command. See Chapter 44, “Performing Interactive and Batch Operations Using ESSCMD” for information about ESSCMD.

## Copying a Database

When you copy a database, all files associated with the database are also copied to the destination application. Before copying, make sure you have enough disk space to contain a full copy of the databases and of the related files.

- To copy a database:
  1. Connect to the server on which the database resides.
  2. From the Application Desktop window, select the application that contains the database.
  3. Select the appropriate database in the selected application.
  4. Select Database > Copy. The following dialog box is displayed:

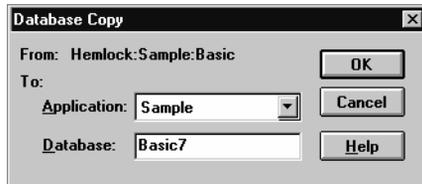


Figure 47-4: Database Copy Dialog Box

5. To copy the database to another database within the same application, in the Database text box type a new database name. Click OK.
6. To copy the database to another database in a different application (the other application must already exist), from the Application list box, select the application name. Then type the name of the new database and click OK.



You can use the COPYDB command in ESSCMD to perform this task. See the online *Technical Reference* in the DOCS directory for information about this command. See Chapter 44, “Performing Interactive and Batch Operations Using ESSCMD” for information about ESSCMD.

## Renaming a Database

When you rename a database, the database, the database directory (`ESSBASE\APP\appname\dbname`), and the outline file (`.OTL`) are renamed. All other objects in the database (for example, calc scripts) with the same name as the database are not renamed.

- To rename a database:
  1. Connect to the server on which the application resides.
  2. From the Application Desktop window, select the application that contains the database.
  3. Select the appropriate database in the selected application.
  4. Select Database > Rename. The following dialog box is displayed:

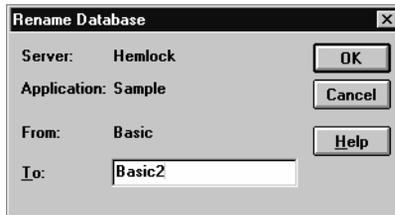


Figure 47-5: Rename Database Dialog Box

5. In the To text box, enter the new name for the database.
6. Click OK. Hyperion Essbase renames the database and the directory under the `\ESSBASE\APP\appname` directory.



You can use the `RENAMEDB` command in `ESSCMD` to perform this task. See the online *Technical Reference* in the `DOCS` directory for information about this command. See Chapter 44, “Performing Interactive and Batch Operations Using `ESSCMD`” for information about `ESSCMD`.

## Deleting a Database

When you delete a database, all objects within the database are also deleted. The `\ESSBASE\APP\appname\dbname` directory and all files located in the directory are deleted.

- To delete a database:
  1. Connect to the server on which the application that contains the database resides.
  2. From the Application Desktop window, select the application that contains the database.
  3. Select the appropriate database in the selected application.
  4. Select Database > Delete. The following dialog box is displayed:

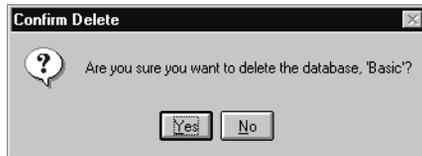


Figure 47-6: Confirm Delete Dialog Box for Databases

5. Click Yes or No.



You can use the `DELETEDB` command in `ESSCMD` to perform this task. See the online *Technical Reference* in the `DOCS` directory for information about this command. See Chapter 44, “Performing Interactive and Batch Operations Using `ESSCMD`” for information about `ESSCMD`.

## Copying Objects

Objects related to databases or applications are outlines, calc scripts, report scripts, rules files, and data sources. You can copy any of these objects, except outlines, to another application, database, server, or client location.

- To copy an object:
  1. Connect to the server on which the object resides.
  2. From the Application Desktop window, select the application.
  3. Select the appropriate database in the selected application.
  4. In the Application Desktop window, click the button corresponding to the object file.
  5. Select File > Copy. The following dialog box is displayed:

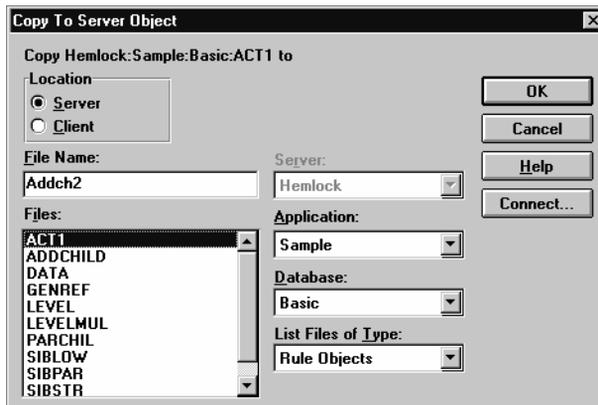


Figure 47-7: Copy to Server Object Dialog Box

6. In the File Name text box, enter the name of the object.
7. Select the server, application, and database names correspond to the destination of the new object.
8. Click OK.

## Renaming Objects

You can rename any object, except outlines. An outline always has the same name as the database, so you need to rename the database to rename the outline.

- To rename an object:
  1. Connect to the server on which the object resides.
  2. From the Application Desktop window, select the application.
  3. Select the appropriate database in the selected application.
  4. In the Application Desktop window, click the button corresponding to the object file.
  5. Select File > Rename. The following dialog box displayed:

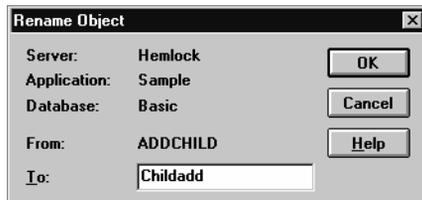


Figure 47-8: Rename Object Dialog Box

6. In the To text box, enter the new name.
7. Click OK.

## Deleting Objects

You can delete any object, except an outline. An outline is a required part of a database, so you need to delete the database to delete the outline.

- To delete an object:
  1. Connect to the server on which the object resides.
  2. From the Application Desktop window, select the application.
  3. Select the appropriate database in the selected application.

4. In the Application Desktop window, click the button corresponding to the object file.
5. Select File > Delete. The following dialog box is displayed:

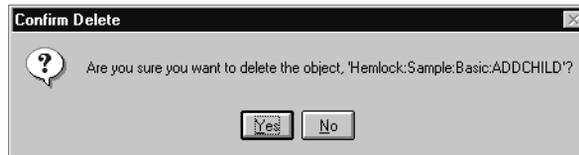


Figure 47-9: Confirm Delete Dialog Box for Objects

6. Click Yes or No.

## Locking and Unlocking Objects

Hyperion Essbase uses a check-out facility for server-based objects to ensure that no more than one user attempts to modify an object at one time. This section describes how to lock and unlock objects.

**Note:** Locking objects is not the same as locking data blocks. The Hyperion Essbase Kernel handles locking for data blocks, but not for objects. See Chapter 42, “Ensuring Data Integrity,” for information about locking data blocks.

### Locking Objects

By default, whenever a user opens a server-based object, Hyperion Essbase locks the object. If a second user attempts to open the same object, a message is displayed. The message indicates that the object is locked and tells the user who locked it.

You can avoid the lock error by unchecking “Lock file” in the Application Desktop window before attempting to open an object. This action essentially opens the file in read-only mode, and you cannot save any changes you make.

## Unlocking Objects

There are two ways to unlock a previously locked object:

- Save the locked file to the server.
- Select File > Unlock. This action unlocks a file that you previously locked but did not save to the server.

**Note:** Users can unlock only objects that they have locked themselves. A user with supervisor privileges can unlock any object.

## Porting Applications Across Platforms

The Hyperion Essbase OLAP Server runs on multiple platforms, including Windows platforms and Unix platforms. For a list of supported platforms and information on how to install and configure the Hyperion Essbase server on each platform, see the *Hyperion Essbase Installation Guide*.

After you install an application, you may want to port the application to a server that may be running a different operating system. This section describes how to port an installed Hyperion Essbase application to another Hyperion Essbase server machine.

Porting Hyperion 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 a Hyperion Essbase application to a server running a different operating system, you need to identify which Hyperion Essbase files are compatible with the new operating system.

The following file types are compatible between operating systems:

- ASCII text files. The Hyperion Essbase ASCII text files are calc scripts (.csc) and report scripts (.rep). Also, data files can be ASCII text files.
- Data load rules files. These files are binary files, but they are compatible between operating systems. Data load 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 need to 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
- Hyperion Essbase kernel files with the extension .esm

**Note:** If you are using the Linked Reporting Objects feature, you need to relink any files or cell notes on the new server. For more information, see Chapter 12, “Linking Objects to Hyperion Essbase Data.”

## Checking File Names

When transferring files to a UNIX system, you need to be aware of the case of file names. UNIX is a case-sensitive operating system, and files are recognized only if they have the correct case. For example, in certain ESSCMD operations, you need to specify a file name, and the file name must be entered with the correct case.

The Hyperion Essbase system files use the following naming conventions on UNIX systems:

- Executable files have no extension and are uppercase (ESSBASE, ESSCMD).
- Static library files have the file extension .a and are in lowercase (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 (`essbase.sec`).
- Message database files have the file extension `.mdb` and are in lowercase (`essbase.mdb`).
- Online help files have the file extension `.hlp` and are in lowercase (`esscmd.hlp`).

Hyperion Essbase files on UNIX systems are capitalized with *proper* case: the first letter is uppercase, and the remaining letters are lowercase. The following table gives examples of names for different file types:

*Table 47-4: File Naming Examples for UNIX*

File Type	Example
Database files	Mydb.db
Data files	Mydb.pag
Index files	Mydb.ind
Outline files	Mydb.otl
Data load rules files	Atlanta.rul
Data files to load	Atlanta.txt
Calc scripts	Mycalc.csc
Report scripts	Myrepo.rep
Archive files	Mydb.arc
Application log files	Myapp.log

**Note:** The application name is an exception to the above rule. The application name can be in lower case.

Table 47-5 lists several examples of valid and invalid file names on UNIX systems:

*Table 47-5: Valid and Invalid File Names on UNIX*

<b>Valid File Names</b>	<b>Invalid File Names</b>
Model.csc	MODEL.CSC
Monthly.rep	Monthly.Rep
Forecast.otl	forecast.otl
Actuals.rul	AcTuAlS.rUl
My_File.txt	My_File.Txt

**Note:** The Hyperion Essbase server does not allow long file names for applications, databases, calc scripts, reports, and other database files. All file names for objects you create must conform to the standard 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 Hyperion Essbase Application Manager to transfer the compatible application files. If the servers are not connected, you need to redefine server information on the new server before reloading the database.

### Using Using FTP to Transfer

Using FTP, you can transfer files directly between operating systems. You should transfer only the files 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, Hyperion Essbase Application Manager can see these files but cannot open them. After you use FTP to transfer files, you should rename the files on the server to ensure that they are capitalized with proper case. Alternatively, you can use FTP to rename the file when you transfer the file:

```
ftp>put oldfile Newfile
```

## Using Hyperion Essbase Application Manager to Transfer

Using Hyperion Essbase Application Manager, you can transfer files from the client machine to the server by using File > Save As. For example, you can connect to a Windows NT server, open an outline, and then save it to a UNIX server. The file types you can transfer from Hyperion Essbase Application Manager are outline files, report scripts, calc scripts, and rules files.

When you save a file from Hyperion Essbase Application Manager to a UNIX server, proper case is applied automatically regardless of what case you type.

## Redefining Server Information

If the server you are porting to is not connected to the existing server, you need to redefine some Hyperion Essbase server information on the new server.

- To redefine server information, follow these steps:
  1. To create users and specify their privileges, use Hyperion Essbase Application Manager on the new server. See Chapter 17, “Managing Security at Global and User Levels,” for help.
  2. To create the applications and databases that you want to port, use Hyperion Essbase Application Manager on the new server. See Chapter 7, “Creating Applications and Databases,” for help.
  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. Make sure the application is not running while you copy these files.
  4. Copy compatible files from the old server to the new server. For help, see “Identifying Compatible Files” on page 47-15.
  5. Reload the database. For help, see “Reloading the Database” on page 47-20.

## 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 need to repopulate the database by reloading the data from a data file and a data load rules file (if applicable). One way you can reload is to export the data to an ASCII text file, transfer the text file to the new server, and then use the text file to load data.

# Backing Up Data and Recovering Databases

---

This chapter tells how to recover a database and how to back up a database and describes other ways that Hyperion Essbase protects data.

This chapter includes the following sections:

- “Backing Up a Database” on page 48-1
- “Recovering from a Crashed Database” on page 48-5

If you are migrating from a previous release of Hyperion Essbase, see the *Hyperion Essbase Start Here* booklet.

## Backing Up a Database

A key part of a database maintenance routine is making regular backups of Hyperion Essbase data. The most common method of backing up is *archiving*. Archiving makes copies of Hyperion Essbase data files exactly as they exist on the system. You can also back up data by *exporting* it. Exporting makes a copy of data in an ASCII text format. This section describes both backup methods.

## Archiving Data

To archive data, use the Hyperion Essbase `BEGINARCHIVE` and `ENDARCHIVE` commands along with a third-party backup utility of your choice. The `BEGINARCHIVE` command places database in read-only mode and provides a list of files to back up. Users can remain logged in, but they are not able to modify data during the archive procedure.

**Note:** If you back up your data without using `BEGINARCHIVE`, make sure that all Hyperion Essbase applications are closed and that all users are logged out during the backup procedure. Otherwise, you risk corrupting the database.

The file list generated by `BEGINARCHIVE` includes linked reporting object files and partitioning database files, if applicable. For more information on Linked Reporting Objects, see Chapter 12, “Linking Objects to Hyperion Essbase Data.” For information on partitioning, see Chapter 6, “Designing Partitioned Applications.”

► To archive data:

1. Issue the `BEGINARCHIVE` command in `ESSCMD`. `BEGINARCHIVE` does the following:
  - Commits any modified data to disk.
  - Switches the database to read-only mode.
  - Reopens the database files in shared, read-only mode.
  - Creates a file containing a list of files that need to be backed up. By default, the file is called `archive.lst`. See the `BEGINARCHIVE` command in the online *Technical Reference* in the `DOCS` directory for more information on `archive.lst`.

If a user tries to modify data during the archive procedure, an error message informs the user that data is in read-only mode for archiving. No user can update the database until the `ENDARCHIVE` command is issued. (See step 3.)

Read-only mode persists until the `ENDARCHIVE` is issued, even if the database is shut down or restarted.

2. Use a third-party backup utility to back up the files listed in `archive.lst`. These files are described in “Recovering from a Crashed Database” on page 48-5.

Be sure to back up data on every disk volume Hyperion Essbase uses. For information about data storage on multiple volumes, see Chapter 40, “Introducing the Hyperion Essbase Kernel.”

3. Complete the archiving process by issuing the `ENDARCHIVE` command in `ESSCMD`. `ENDARCHIVE` does the following:
  - Puts the database into read-write mode.
  - Re-opens database files in exclusive, read-write mode.

**Note:** You must issue an `ENDARCHIVE` command to put the database back into read-write mode, even if you shut down and restart the database. The `ENDARCHIVE` command does not restart the database.

If you try to cancel the archive after issuing `BEGINARCHIVE` and receive a “can’t cancel” message, most likely the system is in the final stage of writing items to the drive and has reached the point where the operation cannot be cancelled.

## Exporting Data

You can back up data by exporting instead of archiving. The `EXPORT` utility copies data to an ASCII text file that you specify; it does not compress data. The exported file contains data only and does not include control, outline, or security information. The maximum size of the exported file is 2 gigabytes.

Archiving is the recommended method of backing up. However, you may use the `EXPORT` utility when doing any of the following:

- Transferring data across platforms
- Backing up only a certain portion of the data; for example, level 0 blocks
- Creating an exported file in text format, rather than binary format

**Note:** You can also export data by creating reports. For more information, see Chapter 39, “Copying Data Subsets and Exporting Data to Other Programs.”

## Using the EXPORT Utility

To export data, select Database > Export in Hyperion Essbase Application Manager or issue EXPORT in ESSCMD. For ESSCMD syntax information, see the online *Technical Reference* in the DOCS directory.

You can use the EXPORT utility to specify the amount of data to export:

- All data
- Level 0 blocks only (blocks containing only level 0 sparse member combinations)
- Data from input blocks only (blocks containing data from a data load)

## Reloading Exported Data

When you reload data that has been exported, Hyperion Essbase marks the data as input data. If you reload data exported from level 0 blocks or input blocks, you must recalculate the database after reloading. When you recalculate the database, Hyperion Essbase recalculates every data block.

If you export *all* data in a database and then reload, Hyperion Essbase marks all blocks in the database as input blocks. Consequently, if you try to clear data by selecting Database > Clear Data > Non-Input Blocks from Hyperion Essbase Application Manager, no data is cleared because the database contains no non-input blocks.

**Note:** To clean the status of the blocks in a database after exporting all data and reloading, you can run the following calc script:

```
Set ClearUpdateStatus Only;  
Calc All;
```

## Recovering from a Crashed Database

After a server interruption such as a crash, Hyperion Essbase recovers your database, rolling back all transactions that were active when the interruption occurred. Hyperion Essbase also recovers and consolidates free fragments (unused addressable units in the data blocks).

Hyperion Essbase recovers data as soon as the server is started after a server interruption. Recovery consists of the following 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 freespace recovery rebuilds the data freespace tables. The size of the index determines the duration of this phase.

Only a media failure (faulty disk, disk failure, or head crash) requires you to restore from archived files. See “Archiving Data” on page 48-2.

---

**CAUTION:** Do not move, copy, modify, or delete any of the following files: `ESSxxxxx.IND`, `ESSxxxxx.PAG`, `tablename.IND`, `tablename.ESM`, `tablename.TCT`. Doing so can result in data corruption.

---

## What to Expect If a Server Interruption Occurs

The following table lists types of server interruptions and their results:

*Table 48-1: Hyperion Essbase Recovery Handling*

Type of Interruption	Result
<ul style="list-style-type: none"> <li>• Power loss on server machine</li> <li>• Operating system crash</li> <li>• Server stopped with Ctrl + C keys</li> </ul>	Server stops. When you restart the server, Hyperion Essbase recovers the database.
<ul style="list-style-type: none"> <li>• Operation cannot complete due to system limitations</li> <li>• Memory shortage</li> <li>• Out of disk space</li> <li>• Application stopped with Ctrl+C keys</li> </ul>	Hyperion Essbase performs fatal error handling. You may need to allocate more memory or disk space and restart the server.
Server crash	Hyperion Essbase Exception Manager creates error log file of type .XCP. Server stops. When you restart the server, Hyperion Essbase recovers the database.

Table 48-2 shows what you need to do if a server interruption occurs during a transaction. How Hyperion Essbase recovers from an interruption depends on the transaction isolation level setting (committed or uncommitted access). See Chapter 42, “Ensuring Data Integrity,” for more information.

*Table 48-2: Recovery Procedures for Server Requests*

Type of Lock	Recommended Action
Lock (for spreadsheet update)	Issue the lock command again.
Send (spreadsheet update)	If Hyperion Essbase issues an error, repeat the last send operation.
Calculation	Repeat the last calculation. See Part V, “Calculating Data.”
Data load	Repeat the last data load (see Chapter 23, “Performing a Data Load”) or load the error log file (see Chapter 24, “Debugging and Optimizing Data Loads”).
Arithmetic data load (adding to or subtracting from values in the database)	<p>Look in the application log file to ascertain the last row for which Hyperion Essbase logged a completion message. Repeat the data load from the row that immediately follows the last row for which Hyperion Essbase logged the completion message.</p> <p>See Chapter 23, “Performing a Data Load” and Chapter 24, “Debugging and Optimizing Data Loads.”</p>
Restructure	Repeat the last operation that caused a restructure. Check for temporary files and delete them.

**Note:** If the UPDATECALC parameter is set to FALSE, Hyperion Essbase recalculates the entire database if an interruption occurs during a calculation. (The parameter’s default value is TRUE.) For more information on UPDATECALC, see the online *Technical Reference* in the DOCS directory.

## Spreadsheet Update Logging

For extra protection against data loss and for spreadsheet audit information, Hyperion Essbase provides a spreadsheet update logging facility. Enable this facility by using the SSAUDIT or SSAUDITR parameter in the `ESSBASE.CFG` server file. For information on the `ESSBASE.CFG` file, see the online *Technical Reference* in the `DOCS` directory.

SSAUDITR clears update log files automatically after archiving, when you issue the `ENDARCHIVE` command. SSAUDIT does not remove log files after archiving. See “Archiving Data” on page 48-2 for information on the `ENDARCHIVE` command.

You can specify SSAUDIT for all databases on the server or for individual databases. See the online *Technical Reference* in the `DOCS` directory for syntax information.

Hyperion Essbase handles recovery under normal situations. However, sometimes you may want to load the spreadsheet update log file manually. For example, if you have restored from a recent backup and do not want to lose changes made since the backup was made or you experience a media failure, you can recover transactions from the update log file. To do so, use the Hyperion Essbase command-line facility, `ESSCMD`, from the server console.

The following `ESSCMD` command sequence loads the update log file:

```
LOGIN hostnode username password
SELECT application_name database_name
LOADDATA 3 filepath:application_name.ATX
EXIT
```

To simplify the process of loading the update log file, prepare a batch file as described in Chapter 44, “Performing Interactive and Batch Operations Using `ESSCMD`.”

**Note:** When SSAUDIT or SSAUDITR is specified, Hyperion Essbase logs spreadsheet update transactions chronologically. Hyperion Essbase uses two files:

`audit_database_name.ATX` stores each update transaction as a unit.

`audit_database_name.ALG` contains historical information: date, time, size, and starting row of each transaction.

The spreadsheet update log can get quite large; even if you are using SSAUDITR, Hyperion Essbase clears the log only after you archive data. If spreadsheet update activities are frequent in your application, you may want to manually delete the log file periodically.

When a database is started after a shutdown, if spreadsheet logging is enabled, Hyperion Essbase writes the following message to the database log file:

```
Starting Spreadsheet Log
volumename\application_directory\application_name\
database_name\database_name.atx For Database database_name
```

For example:

```
Starting Spreadsheet Log \ESSBASE\app\appl\sample\sample.atx for
database sample
```

To ensure successful spreadsheet update logging, stop and restart your application after either of the following:

- Any operation that causes a restructure. See “Database Restructuring” on page 40-14.
- Running any of the following database commands:

```
CREATEAPP
CREATEDB
COPYDB
RENAMEDB
```

Hyperion Essbase ensures that if you enable spreadsheet logging, updates cannot take place without being logged. If Hyperion Essbase cannot write to the update log for any reason, Hyperion Essbase stops the transaction and issues an error message.

SSAUDIT and SSAUDITR are available only from the ESSBASE.CFG file.



# Performance-Related Settings

You can customize Hyperion Essbase at the database level, using either the Database Settings dialog box in Hyperion Essbase Application Manager or ESSCMD.

This section contains information on many performance-related Hyperion Essbase settings. It focuses on general performance rather than, for example, Calculator, data load, or Report Writer performance.

For information on optimizing Calculator performance, see the following chapters:

- Chapter 33, “Optimizing Calculations”
- Chapter 34, “Using Intelligent Calculation to Optimize Calculation”

For information on optimizing data loads, see Chapter 24, “Debugging and Optimizing Data Loads.”

For information on optimizing Report Writer performance, see Chapter 38, “Optimizing Your Reports.”

This appendix provides general-purpose information and does not account for the wide variety of configuration possibilities. For more detailed information on the Hyperion Essbase kernel, see the following chapters:

- Chapter 15, “Estimating Disk and Memory Requirements for a Database”
- Chapter 40, “Introducing the Hyperion Essbase Kernel”
- Chapter 41, “Specifying Hyperion Essbase Kernel Settings”
- Chapter 42, “Ensuring Data Integrity”

**Note:** If you are migrating your database, see the *Hyperion Essbase Start Here* booklet for information about the default settings after migration.

Table A-1 helps you understand what each setting controls and where to adjust each setting:

*Table A-1: Database Settings and Defaults*

<b>Setting</b>	<b>Hyperion Essbase Application Manager Setting</b>	<b>ESSCMD Command</b>	<b>Default Value and Comments</b>
Index cache size	Database > Settings, Storage tab	SETDBSTATEITEM 12	10 MB (default), 1 MB (minimum)  Recommendation: Combined size of all ESS*.IND files if possible; as large as possible otherwise. See “Sizing Your Index, Data File, and Data Caches” on page 15-15.
Data file cache size	Database > Settings, Storage tab	SETDBSTATEITEM 27	32 MB (default), 8 MB (minimum)  Recommendation: Combined size of all ESS*.PAG files if possible; as large as possible otherwise. See “Sizing Your Index, Data File, and Data Caches” on page 15-15.
Data cache size	Database > Settings, Storage tab	SETDBSTATEITEM 5	3 MB (default and minimum) Recommendation: 0.125 * data file cache size value. However, certain calculations require a larger data cache size. In general, if you have to choose between allocating memory to the data file cache and allocating memory to the data cache, choose the data file cache. See “Sizing Your Index, Data File, and Data Caches” on page 15-15.

Table A-1: Database Settings and Defaults (Continued)

Setting	Hyperion Essbase Application Manager Setting	ESSCMD Command	Default Value and Comments
Index page size	Database > Settings, Storage tab	SETDBSTATEITEM 13	<p>With direct I/O, 8 KB (default). With buffered I/O, 1KB (default and minimum).</p> <p>Recommendations:</p> <p>If direct I/O is in effect, Hyperion Essbase ignores the index page size setting and always uses an index page size of 8 KB.</p> <p>If buffered I/O is in effect, use the default setting. Hyperion Essbase accepts any value between 1 KB and 8 KB, but only uses values of 1 KB, 2 KB, 4 KB, or 8 KB. Hyperion Essbase rounds up the value you enter to an acceptable value if necessary.</p>
Cache memory locking	Database > Settings, Storage tab	SETDBSTATEITEM 26	<p>Disabled.</p> <p>Recommendation: If you want to give the Hyperion Essbase kernel priority usage of the system RAM, then enable cache memory locking, being careful to leave at least one third of the system RAM for operations other than Hyperion Essbase kernel. Otherwise, disable cache memory locking.</p>
Isolation level	Database > Settings, Transaction tab	SETDBSTATEITEM 18	<p>Uncommitted.</p> <p>See Chapter 42, “Ensuring Data Integrity,” for more information.</p>

Table A-1: Database Settings and Defaults (Continued)

Setting	Hyperion Essbase Application Manager Setting	ESSCMD Command	Default Value and Comments
Commit Blocks	Database > Settings, Transaction tab	SETDBSTATEITEM 21	3,000  Effective only when isolation level is uncommitted. <b>Note:</b> <ul style="list-style-type: none"> <li>• A zero value means no synchronization points; Hyperion Essbase commits all affected blocks at the end of a transaction.</li> <li>• If either Commit Blocks or Commit Rows has a non-zero 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.</li> </ul>
Commit Rows	Database > Settings, Transaction tab	SETDBSTATEITEM 22	0 (no implicit commit); A synchronization point occurs at the end of a transaction.  Effective only when isolation level is uncommitted.  See the note under Commit Blocks.
Wait for write access to locked data block	Database > Settings, Transaction tab	SETDBSTATEITEM 20	Indefinite wait (-1 in ESSCMD).  Effective only when isolation level is committed.
Pre-image access	Database > Settings, Transaction tab	SETDBSTATEITEM 19	Enabled.  Effective only when isolation level is committed.

Table A-1: Database Settings and Defaults (Continued)

Setting	Hyperion Essbase Application Manager Setting	ESSCMD Command	Default Value and Comments
Disk Volumes: volume name	Database > Settings, Storage tab	SETDBSTATEITEM 23 SETDBSTATEITEM 24	If you do not specify this setting, the Hyperion Essbase kernel uses only the volume that ARBORPATH points to and fills the entire volume as needed.  Replaces DISKVOLUMES <i>volume_name</i> in ESSBASE.CFG, which is then used only for initial migration.
Disk volumes: partition size	Database > Settings, Storage tab	SETDBSTATEITEM 23 SETDBSTATEITEM 24	Unlimited. (Uses all available space on the specified volume.)
Disk volumes: file type	Database > Settings, Storage tab	SETDBSTATEITEM 23	Index and data.
Disk volumes: maximum file size	Database > Settings, Storage tab	SETDBSTATEITEM 23	2 GB
Retrieval buffer size	Database > Settings, General tab	SETDBSTATEITEM 16	10,240 B (10 KB).  <b>Note:</b> In Version 4, this setting is specified as REPTKBYTEBUF in ESSBASE.CFG.
Retrieval sort buffer size	Database > Settings, General tab	SETDBSTATEITEM 17	10,240 B (10 KB).  <b>Note:</b> In Version 4, this setting is specified as REPTKBYTESORTBUF in ESSBASE.CFG.

Table A-1: Database Settings and Defaults (Continued)

<b>Setting</b>	<b>Hyperion Essbase Application Manager Setting</b>	<b>ESSCMD Command</b>	<b>Default Value and Comments</b>
Data compression	Database > Settings, Storage tab	SETDBSTATEITEM 14 SETDBSTATEITEM 15	<p>Bitmap compression enabled.</p> <p>In Hyperion Essbase Application Manager, you use a single setting to choose bitmap compression, RLE (run-length encoding) compression, or no compression.</p> <p>In ESSCMD, use SETDBSTATEITEM 14 to enable or disable compression and SETDBSTATEITEM 15 to control compression type.</p>

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

**administrator.** An individual who installs and maintains the Hyperion Essbase system including setting up user accounts and security. *See also* database administrator, system administrator.

**Advanced Interpretation mode.** An option in Hyperion Essbase Spreadsheet Add-in that you use to define a layout through drill through or Hyperion Essbase Query Designer or by typing data into the sheet. When you construct a free-form report in Advanced Interpretation mode, Hyperion Essbase interprets the member names and creates a default view that is based on the location of the labels.

**agent.** A process on the server that starts and stops applications and databases, manages connections from users, and handles user-access security. Referred to as ESSBASE.EXE.

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

**aggregate.** *See* consolidate.

**alias.** An alternate name for a dimension, member, or description.

**alias table.** A table that stores aliases for the dimensions or members in a database.

**ancestor.** A branch member that has members below it. For example, in the Sample Basic database, Qtr2 and Year are ancestors of April.

**application.** A management structure containing one or more Hyperion Essbase databases and the related files that control many system variables, such as memory allocation and autoloading parameters.

**application designer.** An individual who designs, creates, and maintains Hyperion Essbase applications and databases.

**application log file.** A record of user actions that are performed on an application.

**Application Manager.** Hyperion Essbase software that you use to create and maintain Essbase applications.

**Application Programming Interface.** (API) A library of functions that you can use in a custom C or Visual Basic program to access the Hyperion Essbase server.

**application server.** A middle-tier server that is used to run application processes.

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

**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.

**asymmetric report.** A report characterized by groups of members that differ by at least one member across the groups. There can be a difference in the number of members or the names of members. For example, a report based on Sample Basic can have three members grouped under “East” and two members grouped under “West.”

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

**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 the following calculation operations on groups of members: Sum, Count, Avg, Min, Max. The calculation is based on the attributes associated with the members. This dimension is calculated dynamically and is not visible in the database outline. For example, by using the Count member, you can calculate how many Red products had sales in New York in January. By 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 that is based on the attributes or qualities of the members of its base dimension.

**attribute reporting.** A process of defining reports that is based on the attributes of the base members in the database outline.

**attribute type.** 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. For example, because the Ounces attribute dimension has the type numeric, you can use the number of ounces that is specified as the attribute of each product to calculate the profit per ounce for that product.

**bang character.** (!) A character that terminates a series of report commands and requests information from the database. A report script must be terminated with a bang character; several bang characters may be used within a report script.

**base currency.** The currency in which daily business transactions are performed.

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

**batch calculation.** Any calculation on a database that is done in batch; for example, a calc script or a full database calculation. Dynamic calculations are not considered to be batch calculations.

**batch file.** Operating system file that can call multiple ESSCMD scripts and run multiple sessions of ESSCMD. Batch files handle batch data loads and complex calculations, and can include commands that run report scripts. You can run a batch file on the server from the operating system prompt. On Windows systems, batch files have .BAT file extensions. On UNIX, a batch file is written as a shell script.

**batch processing mode.** A method of using ESSCMD to write a batch or script file that can be used to automate routine server maintenance and diagnostic tasks. ESSCMD script files can execute multiple commands and can be run from the operating system command line or from within operating system batch files. Batch files can be used to call multiple ESSCMD scripts or run multiple instances of ESSCMD.

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

**build method.** A method used to modify database outlines. You choose a build method based on the format of data in data source files.

**cache.** A buffer in memory that holds data temporarily.

**calc script.** *See* calculation script.

**calculation.** The process of aggregating or of running a calc script on a database.

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

**cascade.** The process of creating multiple reports for a subset of member values.

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

**cell note.** A text annotation of up to 599 characters for a cell in a Hyperion Essbase database. Cell notes are a type of linked reporting object.

**change log.** *See* outline change log.

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

**clean block.** A data block is marked as clean if the database is fully calculated, if a calc script calculates all dimensions at once, or if the SET CLEARUPDATESTATUS command is used in a calc script.

**client.** A client interface, such as the Hyperion Essbase Spreadsheet Add-in software, a custom API program, or Hyperion Essbase Application Manager. A client is also a workstation that is connected to a server through a local area network.

**client log file.** A record of all messages, actions, and errors that are generated by a client.

**column.** A vertical display of information in a grid or table. A column can contain data from a single field, derived data from a calculation, or textual information. The words column and field are sometimes used interchangeably. Contrast with row.

**column heading.** A part of a report that lists members across a page. When you can define columns that report on data from more than one dimension, you produce nested column headings. A member that is listed in a column heading is an attribute of all data values in its column.

**committed access.** A Hyperion Essbase kernel Isolation Level setting that affects how Hyperion Essbase handles transactions. Under committed access, concurrent transactions hold long-term write locks and yield predictable results.

**consolidate.** The process of gathering data from dependent entities and aggregating the data up to parent entities. After you enter or load data into dependent child entities, you perform a consolidation to aggregate the data through the organization. As data consolidates, intercompany processing, conversion methods, equity adjustments, and minority ownerships perform calculations on the data. For example, if the dimension Year consists of the members Qtr1, Qtr2, Qtr3, and Qtr4, its consolidation is Year. The terms aggregate and roll-up also describe the consolidation process.

**crosstab reporting.** A type of reporting that categorizes and summarizes data in a table format. The cells within the table 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.

**currency.** A monetary unit of measure that is associated with a balance or transaction.

**currency conversion.** A process that converts currency values in a database from one currency into another currency. To convert one US dollar into the euro, the current exchange rate, for example, 0.923702, is multiplied to the dollar. After conversion ( $1 * 0.923702$ ), the euro amount is .92.

**currency partition.** A dimension type that separates local currency members for a base currency as defined in an application. A currency partition identifies currency types, such as Actual, Budget, and Forecast.

**currency symbol.** A character that represents a currency. For example, the currency symbol for the US dollar is \$ and the currency symbol for the British pound is £.

**data cache.** A buffer in memory that holds uncompressed data blocks.

**data cell.** *See* cell.

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

**data file cache.** A buffer in memory that holds compressed data (.PAG) files.

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

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

**data source.** External data, such as a text file, spreadsheet file, or SQL database that will be loaded into a Hyperion Essbase database.

**data value.** *See* cell.

**database.** A repository of data within Hyperion Essbase that contains a multidimensional data storage array. Each database consists of a storage structure definition (a database outline), data, security definitions, and optional calculation scripts, report scripts, and data loading scripts. An application contains one or more databases.

**database administrator.** An individual who administers a Hyperion Essbase database and may also design, maintain, and create the database.

**database designer.** In Hyperion Essbase, the highest type of access that can be assigned globally (per database). This type of access allows complete calculate and update access and the ability to run report and calc scripts.

**database filter layer.** A layer in the Hyperion Essbase security plan that defines specific settings for database members down to the cell level.

**dense dimension.** A dimension with a high probability that data exists for every combination of dimension members.

**descendant.** Any member below a parent in the database outline. For example, in the Sample Basic database, Qtr2 and April are descendants of Year.

**dimension.** A data category that is used to organize business data for retrieval and consolidation of values. Each dimension usually contains a hierarchy of related members grouped within it. For example, a Year dimension often includes members for each time period, such as quarters and months. Other common business dimensions might be Measure, Natural Accounts, Product, and Market.

**dimension build rules.** Specifications, similar to data load rules, that Hyperion Essbase uses to modify an outline. The modification is based on data in an external data source file.

**dimension type.** A dimension property that enables the use of predefined functionality. Dimensions that are tagged time have a predefined calendar functionality.

**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, have been updated).

**disabled user name.** A user name that has become inactive, meaning that the user is not be able to log on to the server. Users with supervisor privilege can disable a user name for any reason. User names are disabled automatically if they exceed server-specific limitations on login attempts or number of inactive days. Only users with supervisor privilege can enable disabled user names.

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

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

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

**dynamic reference.** A pointer in the rules file to header records in a data source. Header records define data load or dimension build criteria for the fields in a data source.

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

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

**EssCell.** The Hyperion Essbase cell retrieve function. An EssCell function is entered into a cell in Hyperion Essbase Spreadsheet Add-in to retrieve a single database value that represents an intersection of specific database members.

**ESSCMD.** A command-line interface that is used to perform server operations interactively or through a batch file.

**ESSCMD script file.** A text file that contains ESSCMD commands, which Hyperion Essbase executes in order to the end of the file. You can run a script file from the operating system command line or from within an operating system batch file. The default extension is .SCR.

**extraction command.** A type of reporting command that handles the selection, orientation, grouping, and ordering of raw data extracted from a database. These commands begin with the less than (<) character.

**field.** A value or item in a data source file that will be loaded into a Hyperion Essbase database.

**file delimiter.** One or more characters, such as commas or tabs, that separate fields in a data source.

**filter.** A method for controlling access to database cells in Hyperion Essbase. A filter is the most detailed level of security, allowing you to define the varying access levels that different users can have to individual database values.

**FlashBack.** A Hyperion Essbase Spreadsheet Add-in command that restores the previous database view. This command is similar to a typical Undo command.

**formula.** A combination of operators and functions as well as dimension names, member names, and numeric constants. Formulas are used to perform specific calculations on members of a database.

**Free-Form mode.** An option in Hyperion Essbase Spreadsheet Add-in that you use to type report script commands in the worksheet to create reports.

**free-form reporting.** A method of creating reports in which you type members of dimensions or report script commands in a worksheet. Free-form reporting is available in both Advanced Interpretation mode and Free-Form mode.

**function.** A predefined routine that returns a value, a range of values, a Boolean value, or a list of database members. The system provides the following categories of functions: mathematical, relationship, financial, member set, Boolean, statistical, forecasting, allocation, and date-time.

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

**generation name.** A unique name that describes a generation.

**global access layer.** A layer in the Hyperion Essbase security system used to define common access settings for applications and databases.

**global report command.** A command that is executed when it occurs in the report script file and that stays in effect until the end of the report file or until another global command replaces it.

**grouping.** A set of members that is selected by a filtering process and that may be treated as a separate aggregate group. This group behaves very much like a parent to all of its specific members, and it supports full calculation logic, including additive and non-additive calculations. For example, you can use the attribute Small to view and work with all members with the attribute Small.

**header record.** One or more records at the top of a data source. Header records describe the contents of the data source.

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

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

**index.** A method that Hyperion Essbase uses to retrieve data. The retrieval is based on the combinations of sparse dimensions. The term index also refers to the index file.

**index cache.** A buffer in memory that holds index pages.

**index entry.** A pointer to an intersection of sparse dimensions. Each index entry points to a data block on disk and locates a particular cell within the block by means of an offset.

**index file.** A file that Hyperion Essbase uses to store data retrieval information. It resides on disk and contains index pages.

**index page.** A subdivision of an index file containing entries that point to data blocks.

**input block.** A type of data block that has at least one loaded data value.

**input data.** Any data that is loaded from a data source and is not generated by calculating the database.

**intelligent calculation.** A calculation method that tracks which data blocks have been updated since the last calculation.

**interactive mode.** A method of using ESSCMD by entering commands in the ESSCMD window and responding to prompts if necessary. For routine server administration tasks or for complex tasks that require many commands, consider using batch processing mode.

**interdimensional irrelevance.** A situation in which a specific dimension does not intersect with other dimensions. The data is not irrelevant, but because the data in the specific dimension cannot be accessed from the other dimensions, those other dimensions are not relevant to the specific dimension.

**isolation level.** A Hyperion Essbase kernel setting that determines the lock and commit behavior of database operations. Choices are committed access and uncommitted access.

**latest.** A key word that is used within Hyperion Essbase Spreadsheet Add-in or within Report Writer to extract data values based on the member defined as the latest period of time.

**leaf member.** A member that has no children.

**level.** A branch within a dimension. The levels are numbered incrementally from the leaf member (level 0) towards the root.

**level 0 block.** A data block that is created for sparse member combinations when all of the members of the sparse combination are level 0 members.

**level 0 member.** *See* leaf member.

**level name.** A unique name that describes a level.

**linked object.** A term that encompasses linked partitions and linked reporting objects.

**linked partition.** A form of shared partition that provides the ability to use a data cell to link together two different databases. When a user clicks on a linked cell in a worksheet, for example, Hyperion Essbase opens a new sheet displaying the dimensions in the second database. The user can then drill down into the available dimensions in the second database.

**linked reporting object. (LRO)** An external file that is linked to a data cell in a Hyperion Essbase database. Linked reporting objects (LROs) can be cell notes or files that contain text, audio, video, or pictures.

**location alias.** A location alias is a descriptor that identifies a data source. The location alias specifies a server, application, database, user name, and password. Location aliases are set by the database administrator at the database level using Application Manager, ESSCMD, or the API.

**log file.** A system maintained file that records actions and commands. For example, an application log file records user actions that are performed on that application; a client log file records client messages, actions, and errors.

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

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

**member filtering (member selection).** The process of selecting specific members that will be used in a query. You can apply selection criteria such as generation names, level names, pattern match, attributes, and UDAs.

**member select.** A feature within Hyperion Essbase Spreadsheet Add-in that you use to specify members for a report.

**member selection report command.** A type of Report Writer command that selects ranges of members based on database 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 the member to which it is associated and executes the format command before it processes the member.

**Minimum Database Access.** An option group that controls the default security to all of an application's databases, using access settings (such as Read or None) that are applied globally to the application. All users connecting to databases within the application have the access level defined as minimum access; however, individual user privileges may be higher.

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

**multidimensional database.** (MDDDB) A method of referencing data through three or more dimensions. An individual record is the intersection of a point for a set of dimensions.

**multithreading.** A client-server process that enables multiple users to work on the same applications without interfering with each other.

**Named Pipes.** A network protocol stack that enables Hyperion Essbase clients on Windows 95/98 to communicate with Hyperion Essbase servers on Windows NT, when both operating systems use NetBEUI instead of TCP/IP.

**Navigate Without Data.** A Hyperion Essbase Spreadsheet Add-in option that you use to turn off data retrieval. This feature is most useful when a database has Dynamic Calc and Dynamic Calc And Store members.

**nested column headings.** A column heading for a report column that displays data from more than one dimension. For example, in the Sample Basic database, a column heading that contains both Year and Scenario members is a nested column. This is scripted as: <COLUMN (Year, Scenario). The nested column 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.

**numeric attribute range.** A feature that you can use to associate a base dimension member that has a discrete numeric value with an attribute that represents a range of values. For example, to classify your customers by age, you can define an Age Group attribute dimension that contains members for the following age ranges: 0-20, 21-40, 41-60, and 61-80. You can associate each member of the Customer dimension with a particular Age Group range. You can then retrieve data based on the age ranges rather than based on individual age values.

**object.** A file that is related to an application or database. Objects can be outlines, rules files, calc scripts, report scripts, or data sources. They are stored within the application or database subdirectory on the server or client machine.

**on-line analytical processing.** (OLAP) A multidimensional, multi-user, client-server computing environment for users who need to analyze consolidated enterprise data in real time. OLAP systems feature zooming, data pivoting, complex calculations, trend analyses, and modeling.

**outline.** The database structure of a multidimensional database, including all dimensions, members, tags, types, consolidations, and mathematical relationships. Data is stored in the database according to the structure defined in the outline.

**outline change log.** A record of changes made to a Hyperion Essbase database outline.

**page file.** *See* data file.

**page heading.** A type of report heading that lists members that are represented on the current page area of the report. All data values on the page area have the members in the page heading as a common attribute.

**paging.** A storage scheme that makes use of spare disk space by increasing the available memory.

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

**partition area.** A subcube within a database. A partition is composed of one or more areas. These areas are composed of cells from a particular portion of the database. For replicated and transparent partitions, the number of cells within an area must be the same for both the data source and the data target to ensure that the two partitions have the same shape. For example, if the data source area contains 18 cells, the data target area must contain 18 cells. If the data source area contains 18 cells, the data target area must also contain 18 cells to accommodate the number of values

**Partition Manager.** A Hyperion Essbase tool that you use to create and maintain a replicated, linked, or transparent database partition. Partition Manager includes Partition Wizard, a component that contains a series of pages that step you through the partition creation process.

**partitioning.** The process of defining areas of data that are shared or linked between data models. Partitioning can affect the performance and scalability of Hyperion Essbase applications.

**Password Management.** A group of options in the server settings that lets you limit a user's allowed number of login attempts, number of days of inactivity, and number of days using the same password.

**pattern matching.** The ability to match a value with any or all characters of an item that is entered as a criterion. A missing character may be represented by a wild card value 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.

**period.** An interval within the time dimension.

**pivot.** The ability to alter the perspective of retrieved data. When Hyperion Essbase first retrieves a dimension, it expands data into rows. You can then pivot or rearrange the data to obtain a different viewpoint.

**precalculation.** The process of calculating the database prior to user retrieval.

**preserve formulas.** The process of keeping user-created formulas within a worksheet while retrieving new data.

**property.** A characteristic of a member, such as two-pass calculation or shared member. Properties affect how Hyperion Essbase works with the data.

**record.** In a database, a group of fields that make up one complete entry. For example, a record about a customer might contain fields for name, address, telephone number, and sales data.

**redundant data.** Duplicate data blocks that Hyperion Essbase retains during transactions until Hyperion Essbase commits the updated blocks.

**Remove Only.** A Hyperion Essbase Spreadsheet Add-in command that you use to remove only the highlighted cells within a worksheet.

**replicated partition.** A portion of a database, defined through Partition Manager, that you use to propagate as update to data that is mastered at one site to a copy of data that is stored at another site. Users are able to access the data as though it were part of their local database.

**report.** The formatted summary information that is returned from a database after a report script is run. One or more reports can be generated from a report script.

**Report Extractor.** A Hyperion Essbase component that retrieves report data from the Hyperion Essbase database when you run a report script.

**report script.** An ASCII file containing Hyperion Essbase Report Writer commands that generate one or more production reports. Report scripts can be run in batch mode, through the ESSCMD command-line interface, or through Hyperion Essbase Application Manager. The report script is a text file that contains data retrieval, formatting, and output instructions.

**Report Viewer.** A Hyperion Essbase component that displays the complete report after a report script is run. Saved reports typically have the file extension .RPT.

**restore.** An operation to reload data and/or structural information after a database has been damaged or destroyed. Typically performed after you shut down and restart the database.

**restructure.** An operation to regenerate or rebuild the database index and, in some cases, the data files.

**roll-up.** *See* consolidate.

**root member.** The highest member in a dimension branch.

**row.** A horizontal display of information in a grid or table. A row can contain data from a single field, derived data from a calculation, or textual information. The words row and record are sometimes used interchangeably. Contrast with column.

**row heading.** A report heading that lists members down a report page. The members are listed under their respective row names. Rows are indented below a dimension name.

**server.** A multi-user database server that accesses data values based on the intersection of dimension members.

**server application.** *See* server.

**server interruption.** Any occurrence that stops the server, including a crash, a power outage, or a user pressing the Ctrl+C keys.

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

**sibling.** A child member at the same branch level as another child member. For example, in the Sample Basic database, East and West are siblings.

**SMP.** *See* symmetric multiprocessing

**sparse dimension.** A dimension with a low probability that data exists for every combination of dimension members.

**Spreadsheet Add-in.** Hyperion Essbase software that works with your spreadsheet. Hyperion Essbase Spreadsheet Add-in is an add-in module to your spreadsheet software.

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

**stored member.** A member that Hyperion Essbase does not dynamically calculate upon retrieval. Stored members are stored on disk.

**subset.** A cross-section of data. Subsetting further defines members that meet specific conditions.

**substitution variable.** A variable that acts as a global placeholder for information that changes regularly. You set the variable and a corresponding string value; the value can then be changed at any time. Substitution variables can be used in calc scripts, report scripts, Hyperion Essbase Spreadsheet Add-in, and Hyperion Essbase API.

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

**suppress rows.** The option to exclude rows that contain missing values and to underscore characters from spreadsheet reports.

**swapping.** *See* paging.

**symmetric multiprocessing.** (SMP) A server architecture that enables multiprocessing and multithreading. Hyperion Essbase supports multiple threads over SMP servers automatically. Thus, performance is not significantly degraded when a large number of users connect to a Hyperion Essbase server simultaneously.

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

**TCP/IP.** *See* Transmission Control Protocol/Internet Protocol.

**template.** A predefined format that is designed to retrieve particular data on a regular basis and in a consistent format.

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

**time series reporting.** A process of reporting data based on a date calendar (for example, year, quarter, month, or week).

**toolbar.** A series of shortcut buttons providing quick access to commands. Usually located directly below the menu bar. Not all windows have a toolbar.

**Transmission Control Protocol/Internet Protocol.** (TCP/IP) A standard set of communications protocols that are adapted by many companies and institutions around the world and that link computers with different operating systems and internal architectures. You use TCP/IP utilities to exchange files, send mail, and store data to various computers that are connected to local and wide area networks.

**transparent partition.** A form of shared partition that provides the ability to access and manipulate remote data transparently as though it were part of your local database. The remote data is retrieved from the data source each time you request it. Any updates made to the data are written back to the data source and become immediately accessible to both local data target users and transparent data source users.

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

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

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

**uncommitted access.** A Hyperion Essbase kernel setting that affects how Hyperion Essbase handles transactions. Under uncommitted access, concurrent transactions hold short-term write locks and can yield unpredictable results.

**uniform resource locator.** (URL) An address for a resource in the World Wide Web, such as a document, image, downloadable file, service, or electronic mailbox. URLs use a variety of naming schemes and access methods such as HTTP, FTP, and Internet mail. An example of a URL is <http://www.hyperion.com>. A URL can also point to a file on a local or network drive, such as `D:\essbase\docs\index.htm`.

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

**URL.** *See* uniform resource locator.

**user-defined attribute.** *See* UDA.

**validation.** A process of checking a rules file, report script, or partition definition against the outline to make sure the object being checked is valid.

**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 formula; read only data cells; read/write data cells; or linked objects.

**workbook.** An entire spreadsheet file with many worksheets.

# Index

This index spans two volumes. Chapters 1–34 are in Volume I; Chapters 35–48 and an Appendix are in Volume II. Examples of entries: 33-10 means Chapter 33, p. 10; A-2 means Appendix A, p. 2.

## Symbols

- ! (bang) command
  - terminating reports, 35-9
- ! (exclamation points)
  - adding to report scripts, 36-2
  - in names in scripts and formulas, 8-16
- " (double quotation marks)
  - enclosing member names, 20-14, 22-13, 36-3
  - in application and database names, 7-12
  - in dimension and member names, 8-15
  - in ESSCMD commands, 44-2
  - in formulas, 14-11, 26-6, 31-15
  - in header information, 14-24
  - in report scripts, 39-8
  - terms in scripts and formulas, 8-16, 8-17
- #MI values
  - inserting into empty fields, 20-4, 20-16, 22-15
  - instead of #MISSING, 24-12
- #MISSING values, 4-7, 33-36, 37-56
  - aggregating
    - defaults for, 33-36
    - effects on calculation order, 28-15, 28-16, 28-18, 28-20
    - setting behavior for, 33-36
  - averages and, 30-5
  - calculations and, 31-13, 31-53, 33-35
  - disabling, 23-17
  - #MISSING values (*Continued*)
    - formatting in reports, 36-14
    - in calculations, 33-35
    - inserting into empty fields, 20-4, 20-16
    - parents and, 30-3, 33-38
    - replacing with text, 36-25
    - reporting samples, 37-2, 37-4
    - skipping, 9-7, 30-5
    - sorting data with, 36-49
    - specifying in data source, 14-25
    - testing for, 27-5
    - viewing with Personal Essbase, 39-12
- #NOACCESS value, 17-29
- \$ fields, 20-4
- \$ALT\_NAME setting, 11-3
- % (percent signs)
  - as codes in data source, 14-25
  - in names in scripts and formulas, 8-17
  - in report scripts, 37-23
- % operators
  - defining member consolidations, 9-17
  - in calc scripts, 31-39
  - in mathematical operations, 26-16
  - in unary operations, 5-27, 28-5, 28-7
- & (ampersands)
  - in calc scripts, 26-48, 31-52
  - in names, 20-5
  - in names in scripts and formulas, 8-16
  - in report scripts, 36-34
  - & commands, 26-48, 31-52

- () (parentheses)
  - in calc scripts, 26-29
  - in dimension and member names, 8-15
  - in formulas, 31-51
  - in names in scripts and formulas, 8-17, 36-3
  - indicating negative numeric values in fields, 20-4
  - report scripts, 36-33, 37-58
- \* (asterisks)
  - as codes in data source, 14-25
  - in application and database names, 7-12
  - in names in scripts and formulas, 8-16, 36-3
  - options in reports, 37-26
  - used as wildcard, 36-38
- \* operators, 5-27, 9-17, 28-5, 28-7
- + (plus signs)
  - as codes in data source, 14-25
  - in application and database names, 7-12
  - in dimension and member names, 8-15
  - in member names, 20-5
  - in names in scripts and formulas, 8-17, 36-3
- + operators
  - defining member consolidations, 9-17
  - in mathematical operations, 26-16
  - in unary operations, 5-27, 28-5, 28-7
  - member consolidation, 5-26
- , (commas)
  - as file delimiter, 20-6
  - displaying in reports, 36-26
  - in application and database names, 7-12
  - in data fields, 20-4
  - in dimension and member names, 8-15
  - in formulas, 26-35
  - in header records, 21-13
  - in member combinations, 22-25
  - in names in scripts and formulas, 8-16, 36-3
  - suppressing in reports, 36-14, 36-23
- . (periods)
  - in application and database names, 7-12
  - in dimension and member names, 8-15
  - in names in scripts and formulas, 8-17
- / (slashes)
  - as codes in data source, 14-25
  - in names in scripts and formulas, 8-17, 36-3
- / operators
  - defining member consolidations, 9-17
  - in calc scripts, 31-39
  - in mathematical operations, 26-16
  - in unary operations, 5-27, 28-5, 28-7
- /\* \*/ character pairs, 9-27, 31-15
- // (double slashes) in report scripts, 36-3
- : (colons)
  - in application and database names, 7-12
  - in formulas, 26-35
  - in names in scripts and formulas, 8-16, 36-3
- :: (double colons) in formulas, 26-35
- ; (semicolons)
  - in application and database names, 7-12
  - in calc scripts, 31-15, 31-17, 31-36, 31-40
  - in ESSCMD syntax, 44-2
  - in formulas, 26-6, 26-7, 26-16
  - in names in scripts and formulas, 8-17, 36-3
- < (less than signs)
  - in application and database names, 7-12
  - in dimension and member names, 8-15
  - in names in scripts and formulas, 8-17, 36-3
  - in report scripts, 35-9
- = (equal signs)
  - in application and database names, 7-12
  - in calc scripts, 31-39
  - in dimension and member names, 8-15
  - in formulas, 26-15
  - in names in scripts and formulas, 8-16
  - in report scripts, 36-3
- > (greater than signs)
  - in application and database names, 7-12
  - in names in scripts and formulas, 8-17
- > operators, 4-7
  - in formulas, 33-11
  - inserting in calc scripts, 31-40
  - inserting in formulas, 26-3, 26-16
  - overview, 26-46, 33-10
  - usage examples, 3-10, 26-47
- ? (question marks)
  - in application and database names, 7-12
  - used as wildcard, 36-38
- @ (at signs)
  - in dimension and member names, 8-15
  - in names in scripts and formulas, 8-16, 36-3

- @ABS function, 26-38
- @ACCUM function, 26-45, 27-3
- @ALLANCESTORS function, 26-36
- @ALLOCATE function, 26-42, 32-9
- @ANCEST function, 26-36
- @ANCESTORS function, 26-36
- @ANCESTVAL function, 26-44
- @ATTRIBUTE function, 10-37, 26-38
- @ATTRIBUTEVAL function, 10-37, 26-44, 27-6
- @AVG function, 26-38
- @AVGRANGE function, 26-43, 27-3
- @CHILDREN function, 26-36
- @COMPOUND function, 26-45
- @COMPOUNDGROWTH function, 26-45
- @CORRELATION function, 26-41
- @COUNT function, 26-41
- @CURGEN function, 26-44
- @CURLEV function, 26-44
- @CURRMBR function, 26-36
- @CURRMBRRANGE function, 26-36
- @DECLINE function, 26-45
- @DESCENDANTS function, 16-9, 26-36
- @DISCOUNT function, 26-45
- @FACTORIAL function, 26-38
- @GEN function, 26-44
- @GENMBRS function, 26-37
- @GROWTH function, 26-45
- @IALLANCESTORS function, 26-36
- @IANCESTORS function, 26-36
- @ICHILDREN function, 26-36
- @IDESCENDANTS function, 26-36, 31-55
- @ILSIBLINGS function, 26-37
- @INT function, 26-38
- @INTEREST function, 26-45
- @IRR function, 26-45
- @IRSIBLINGS function, 26-37
- @ISACCTYPE function, 26-29
- @ISANCEST function, 26-29
- @ISCHILD function, 26-29
- @ISDESC function, 26-30
- @ISDESCENDANTS function, 22-25
- @ISGEN function, 26-30
- @ISIANCEST function, 26-29
- @ISIBLINGS function, 26-37
- @ISICHILD function, 26-30
- @ISIDESC function, 26-30
- @ISIPARENT function, 26-30
- @ISISIBLING function, 26-30
- @ISLEV function, 26-30
- @ISMBR function, 26-30, 26-48
- @ISPARENT function, 26-30
- @ISSAMEGEN function, 26-30
- @ISSAMELEV function, 26-30
- @ISSIBLING function, 26-30
- @ISUDA function, 26-30
- @LEV function, 26-44
- @LEVMBR function, 26-37
- @LIST function, 26-37
- @LSIBLINGS function, 26-37
- @MATCH function, 26-37
- @MAX function, 26-38
- @MAXRANGE function, 26-43
- @MDALLOCATE function, 26-42, 32-13
- @MDANCESTVAL function, 26-44
- @MDPARENTVAL function, 26-45
- @MDSHIFT function, 26-44
- @MEDIAN function, 26-41
- @MERGE function, 26-37
- @MIN function, 26-38
- @MINRANGE function, 26-43
- @MOD function, 26-38
- @MODE function, 26-41
- @MOVAVG function, 26-42
- @MOVMAX function, 26-42
- @MOVMEAN function, 26-42
- @MOVMIN function, 26-42
- @NEXT function, 26-44
- @NPV function, 26-46
- @PARENT function, 26-37
- @PARENTVAL function, 26-44, 32-7
- @POWER function, 26-38
- @PRIOR function, 26-44, 27-4
- @PTD function, 26-46, 27-1, 30-7
- @RANGE function, 26-37
- @RANK function, 26-41
- @RELATIVE function, 26-37
- @REMAINDER function, 26-38
- @REMOVE function, 26-37
- @ROUND function, 26-38
- @RSIBLINGS function, 26-37
- @SANCESTVAL function, 26-44
- @SHIFT function, 26-44

- @SIBLINGS function, 26-37
- @SLN function, 26-46
- @SPARENTVAL function, 26-45
- @SPLINE function, 26-43
- @STDEV function, 26-41
- @STDEVP function, 26-41
- @STDEV RANGE function, 26-41
- @SUM function, 26-38
- @SUMRANGE function, 26-43
- @SYD function, 26-46
- @TODATE function, 10-37, 26-46
- @TREND function, 26-43, 32-21
- @TRUNCATE function, 26-39
- @UDA function, 26-37
- @VAR function, 5-31, 9-9, 26-39
- @VARIANCE function, 26-41
- @VARIANCEP function, 26-41
- @VARPER function, 5-31, 9-9, 26-39
- @WITHATTR function, 10-37, 26-38
- @XREF function, 7-20, 26-45
- [] (brackets)
  - in application and database names, 7-12
  - in names in scripts and formulas, 8-16, 36-3
- \ (backslashes)
  - in application and database names, 7-12
  - in dimension and member names, 8-15
  - in formulas, 14-11
  - in names in scripts and formulas, 8-16
- \_ (underscores)
  - converting spaces to, 20-11, 22-19
  - in dimension and member names, 8-15
  - in report scripts, 36-3
- { } (braces)
  - in dimension and member names, 8-15
  - in names in scripts and formulas, 8-16, 36-3
  - in report scripts, 35-9, 36-2
- | (vertical bars)
  - in application and database names, 7-12
  - in dimension and member names, 8-15
- ~ (tildes)
  - as character in headings, 36-13
  - as codes in data source, 14-25
- ~ operators, 5-27, 9-17, 28-5, 28-7

- (hyphens, dashes, minus signs)
  - as codes in data source, 14-25
  - in data fields, 20-4
  - in dimension and member names, 8-15
  - in member names, 20-5
  - in names in scripts and formulas, 8-16
  - in report scripts, 36-3
- operators
  - defining member consolidations, 9-17
  - in mathematical operations, 26-16
  - in unary operations, 5-27, 28-5, 28-7
- ' (single quotation marks)
  - in application and database names, 7-12
  - in dimension and member names, 8-15

## Numerics

- 0 (zero) values
  - calculating, 33-35
  - excluding in time balances, 14-25
  - formatting in reports, 36-14, 36-23
  - including in time balances, 14-25
  - replacing with labels, 36-25
  - skipping, 9-6, 30-5

## A

- .A files, 47-5
- A, average time balance codes in data source, 14-25
- aborted transactions, 42-12
- aborting data loads, 23-10
- @ABS function, 26-38
- absolute values, 26-38
- access, 1-3, 1-4, 4-2
  - assigning/reassigning user, 17-11, 17-13, 17-22, 17-24
  - cells in blocks, 4-6, 6-23
  - checking information about, 46-2
  - concurrent, 29-15
  - controlling, 6-8, 6-30, 17-1, 18-1, 40-6
  - data sources, 6-10
    - replicated partitions and, 6-10
  - data targets, 6-10, 6-18
  - defining global levels, 17-31, 17-34, 17-35

- access (*Continued*)
  - getting started tips, 2-1
  - global defined, 17-29
  - global levels, 17-30
  - internal structures optimizing, 4-4
  - levels defined in filters, 18-2
  - levels for users, 17-25
  - linked objects, 12-4
  - local, 6-13
  - locked data blocks, 20-24, 33-23
  - maintaining data integrity, 17-37
  - minimum database level, 17-32
  - modifying, 17-22, 17-27, 17-28, 17-31
  - multi-user, 13-46
  - optimizing, 6-2, 6-8
  - outlines, 38-8
  - overriding levels, 17-33
  - partitioned databases, 6-8, 6-9, 6-26
    - troubleshooting, 16-63
  - remote databases, 6-8
  - restricting, 17-37, 17-39
  - Server Agent, 45-2
  - setting database, 17-24
  - simultaneous, 6-10, 6-23
  - size considerations and, 15-3
  - transactions and, 42-2, 42-3
  - unspecified members, 18-19
- Access databases. *See* SQL databases
- Access DBs setting, 17-23, 17-24
- Account Dimension Properties page (Dimension Properties), 14-7
- account properties
  - shared members and, 9-23
- account reporting values, 30-1
- accounts
  - administrators, 6-31, 16-7
  - users, 6-26, 6-32
- accounts dimension
  - calculating first/last values in, 30-3, 30-4, 30-6
  - calculating time period averages, 30-4
  - calculating variance for, 26-39
  - calculation passes for, 28-23
  - calculations on, 5-29, 5-31, 5-34
  - creating, 9-4
- accounts dimension (*Continued*)
  - currency applications, 43-3
  - description, 5-20
  - flipping values in, 22-27
  - setting, 9-2
  - setting member properties in data source, 14-25
  - time balance members in, 9-4
  - two-pass calculations and, 33-28
  - unary operations in, 28-5
  - usage examples, 4-14, 5-28
  - usage overview, 9-3
- accounts tags, 4-8
  - checking for, 26-29
- @ACCUM function, 26-45, 27-3
- accumulation of values, 26-45
- activity logs, 7-4
- actual expense vs. budgeted
  - getting variance, 26-8, 26-39, 32-2
  - setting properties, 9-9
- actual sales reporting samples, 37-2, 37-44
- ad hoc calculations, 36-16
- ad hoc currency reporting, 43-5
- add as sibling build method, 13-31
  - attribute associations, 13-19
- adding
  - See also* building; creating; defining
  - alias tables to outlines, 11-7, 11-9
  - aliases to calc scripts, 31-48
  - aliases to formulas, 26-25
  - comments to calc scripts, 31-15
  - comments to dimensions, 9-27
  - comments to report scripts, 36-3
  - dimensions to outlines, 3-18, 4-8, 4-11, 8-15, 8-18
    - performance considerations, 33-3, 33-8
    - restructuring and, 40-28
  - dynamically calculated members to calculations, 29-22
  - equations to calc scripts, 26-28, 31-37
  - formulas to calc scripts, 31-16, 31-36, 31-38, 31-51
  - formulas to outlines, 26-5
  - formulas to report scripts, 37-32

adding (*Continued*)

- header information, 14-23
  - headers to data sources, 14-23, 14-24, 21-11, 21-13
  - headings to reports, 35-7, 35-29, 36-4, 36-17
  - members, 6-13
    - as children of specified parent, 13-16
    - as siblings of lowest level, 13-15
    - as siblings with matching strings, 13-13
    - build methods, 13-5
    - example for, 3-13, 3-16
    - for currency conversions, 43-22, 43-25
    - guidelines for, 5-27
    - restrictions, 3-5
    - through header information in the data source, 21-13
    - to dimensions, 8-19, 9-23, 13-13, 13-15, 13-16
    - to member fields, 20-14, 20-15
    - to outlines, 8-15, 13-12
  - members to report scripts, 36-27, 37-57
    - in precise combinations, 36-32
    - with common attributes, 36-37
  - operators to formulas, 26-15
  - page breaks to reports, 36-10, 36-23
  - prefixes or suffixes to fields, 22-20
  - records to data sources, 13-5, 13-8, 13-11
  - shared members to outlines, 13-32, 13-42, 13-43
    - caution for placing, 9-23
  - text to calc scripts, 31-39
  - text to formulas, 26-15
  - titles to reports, 36-24
  - values to empty fields, 22-15
  - values to existing values, 22-22
  - variables to formulas, 26-48
  - variables to report scripts, 36-33, 36-35, 36-36, 36-45
- Adding Dimensions dialog box, 40-29
- addition
- consolidation codes in data source, 14-25
  - prerequisite for, 22-23
  - setting member consolidation properties, 9-18
- addition operator (+)
- unary operations, 28-5

## addition operators (+)

- defining member consolidations, 5-26
  - in unary operations, 5-27, 28-7
  - mathematical operations, 26-16
  - member consolidation, 9-17
- ADDUSER command, 17-12
- adjusting buffer size, 38-2, 38-4
- adjusting column length, 36-13
- Admin page (Partition Wizard), 16-7
- administrative accounts, 6-31
  - setting up, 16-7
- administrative requests, 45-13
- administrators, xli
  - controlling partitioned updates, 6-11
  - getting started with Hyperion Essbase, 2-1, 2-2
  - maintenance routines, 2-4
  - maintenance tasks, 46-18, 48-1
  - minimizing downtime, 6-13
  - usage prerequisites, xliii
  - user-management tasks, 17-7, 17-39, 17-41
- .ADW files, 47-4
- AFTER command, 36-26
  - usage examples, 37-25, 37-36
- Agent log file, 45-4, 45-16
- Agent. *See* Server Agent
- agents, 1-4
- AGENTTHREADS setting, 45-15
- aggregation
  - See also* consolidation
  - formulas vs., 33-21
  - missing values
    - calculations and, 31-53, 33-36
    - effects on calculation order, 28-15, 28-16, 28-18, 28-20
    - in calculations, 31-13
    - parents, 33-38
- AIX servers. *See* UNIX platforms
- .ALG files, 47-1
- alias combinations
  - creating, 11-5
  - defined, 11-2
- Alias Combinations page (Member Properties), 11-6

- alias field type
  - in header records, 14-24
  - in rules files, 14-15
  - nulls and, 13-7, 13-10
- alias tables
  - clearing contents, 11-11
  - copying, 11-9
  - creating, 11-3, 11-7
  - described, 11-2
  - importing/exporting, 11-11, 11-13
  - including in report scripts, 36-42
  - introducing, 11-1, 11-3
  - loading, 11-12
  - maximum per outline, 11-7
  - removing from outlines, 11-10
  - renaming, 11-9
  - reserved generation names in, 36-31
  - selecting, 26-25, 31-48
  - setting as current, 11-7, 11-8
  - updating, 14-21
- ALIAS. *See* alias field type
- aliases
  - adding to calc scripts, 31-48
  - adding to formulas, 26-25
  - adding to report scripts, 36-42, 37-20, 37-39
    - caution for, 36-13
  - allowing changes, 14-10
  - build methods and, 13-3
  - creating, 11-3, 11-4
    - attribute dimension build example, 13-26
    - based on member combinations, 11-5
    - caution for, 16-6, 16-21, 16-32
    - guidelines for, 11-2
    - parent/child dimension build example, 13-12
  - defined, 11-1
  - displaying, 8-25, 11-6
    - in Calc Script Editor, 31-48
    - in Formula Editor, 26-25
  - displaying in reports, 36-43, 37-20
  - duplicate generation fields and, 13-33
  - editing names, 11-6
  - entering in fields, 20-5
  - expanding/contracting member list, 11-6
  - introducing, 11-1
- aliases (*Continued*)
  - maximum characters in names, 11-4
  - multiple, 11-2
  - shared members and, 9-23
  - sorting members by, 36-44
  - specifying for Dynamic Time Series members, 30-13
    - with embedded blanks, 11-3
  - @ALLANCESTORS function, 26-36
  - ALLINSAMEDIM command
    - usage, 36-27
  - @ALLOCATE function, 26-42
  - allocation
    - examples, 26-47, 32-7, 32-9, 32-13
    - functions, 26-5
    - storage, 40-4, 40-11
      - example, 41-24
  - Allocation Manager, 40-4
  - Allow Application to Start option, 17-33
  - Allow Association Chgs option, 13-21
  - Allow Commands option, 17-33
  - Allow Connects option, 17-33
  - Allow Formula Changes option, 14-11
  - Allow Moves option, 14-10
  - Allow Property Changes option, 14-10
  - Allow Updates option, 17-33
  - ALLSIBLINGS command
    - usage, 36-27
  - alphabetizing members, 8-21
  - altering. *See* changing; editing
  - alternate hierarchies (in Hyperion MBA). *See* shared members
  - alternate names. *See* aliases
  - ampersands (&)
    - in calc scripts, 26-48, 31-52
    - in names, 20-5
    - in names in scripts and formulas, 8-16
    - in report scripts, 36-34
  - analysis, 3-2
    - defining objectives, 5-6
    - example, 5-11
    - getting started tips, 2-2
    - optimizing, 12-1
    - single-server applications, 5-4, 5-6

- analyzing database design
  - guidelines, 5-10
- @ANCEST function, 26-36
- ancestor/descendant relationships, 3-6
  - defining calculation order for, 28-4
- @ANCESTORS function, 26-36
- ancestors
  - checking for, 26-29
  - currency conversions and, 43-9
  - defined, 3-6
  - getting, 26-36, 26-44
  - new members with no, 13-12
- ANCESTORS command
  - usage, 36-27
- @ANCESTVAL function, 26-44
- anchoring dimensions, 33-16
- AND operator, 36-22
  - select/reject criteria, 21-16, 21-18
- annotating
  - See also* comments
  - data cells, 12-2
  - databases, 7-13
  - partitions, 16-6, 16-21, 16-32
- .APB files, 47-1
- API (Application Programming Interface)
  - complete listing of functions, 45-13
  - introduction to, 1-5
- API directory, 47-5
- API function calls, 36-52
- API functions, 12-3
  - using VBA functions, 23-18
- .APP files, 47-1
- App Access button, 17-11, 17-13, 17-23
- APP directory
  - location on server for applications, 8-12
  - security file information, 17-2
- Application Access dialog box, 17-27
- Application Copy dialog box, 47-6
- Application Designer access setting, 17-22, 17-23, 17-35
- Application Designer privileges, 17-35
- Application Desktop window, 7-7
- application directory, 17-2, 47-5
- application event log
  - calculation time, 31-9
  - deleting, 46-29
  - description of, 46-28
  - dimensions calculated, 31-9
  - last row committed, 22-23
  - viewing, 46-28
- application files, 47-16
- Application Information dialog box, 46-2
- Application Log Viewer. *See* Log Viewer
- Application Manager, 45-13
  - adding dimensions, 8-18
  - adding members, 8-19
  - annotating databases, 7-13
  - as client interface, 1-5
  - caution for clearing data, 12-3
  - changing data compression defaults, 40-10
  - changing Hyperion Essbase kernel settings, 41-5
  - clearing alias tables, 11-11
  - copying
    - alias tables, 11-9
    - applications, 47-6
    - databases, 47-9
    - objects, 47-12
    - outlines, 8-4, 39-5
  - creating
    - alias tables, 11-7
    - applications, 7-10
    - databases, 7-11
    - rules files, 13-2
    - substitution variables, 7-16
  - creating calc scripts, 31-4, 31-19
  - creating reports, 35-2, 35-12
  - creating text files, 39-7
  - defining calculations, 25-9
    - as default, 25-8
  - deleting
    - alias tables, 11-10
    - applications, 47-8
    - databases, 47-11
    - log files, 46-30, 46-33
    - substitution variables, 7-17
  - described, 7-2

Application Manager (*Continued*)

- enabling two-pass calculations, 33-27
- importing/exporting alias tables, 11-12, 11-13
- importing/exporting outlines, 8-11, 8-13
- loading data, 23-1
- maintaining security information, 7-4
- moving members, 8-22
- naming generations and levels, 8-23
- opening
  - outlines, 8-2, 8-8
  - report scripts, 35-14
- overview, 1-1
- removing linked objects, 12-6
- renaming
  - alias tables, 11-9
  - applications, 47-7
  - databases, 47-10
  - objects, 47-13
- replicating data, 6-14
- restructuring partitioned databases, 6-19
- running calc scripts, 31-8, 31-31, 31-32, 31-34
- saving outlines, 8-7, 8-8
- selecting
  - applications, 7-7
  - data sources, 23-3, 23-4, 23-5
  - databases, 7-7
- setting
  - cache size, 41-8, 41-10, 41-11
  - current alias table, 11-8
  - default calculation, 34-3
  - index page size, 41-14
  - isolation levels, 41-17
- sorting members, 8-21
- specifying data compression, 41-27
- specifying disk volumes, 41-22
- specifying file size, 12-5
- starting/stopping applications, 7-5, 45-6, 45-7
- starting/stopping databases, 7-6, 45-8, 45-9
- synchronizing outlines, 16-50
- transferring files from client to server, 47-19
- updating substitution variables, 7-18
- verifying outlines, 8-7

Application Manager (*Continued*)

- viewing
  - and changing user access, 17-27
  - applications and databases, 21-2
  - linked objects, 12-6
  - log files, 46-28, 46-31
  - storage information, 4-8
- Application Programming Interface. *See* API
- application server, 45-13, 45-14
- Application Settings dialog box
  - global application access, 17-32
  - minimum database access settings, 17-32
  - options, 17-33
  - specifying application file size, 12-5
- application startup, results of, 7-3
- Application Tools, 1-2
- applications, 1-3
  - See also* partitioned applications
  - accessing, 1-3, 17-23, 17-31
  - analytical vs. transaction processing, 3-1
  - associating index files with, 46-16
  - associating with calc scripts, 31-19, 31-25, 31-28
  - building, 7-9
  - client-server architecture described, 1-4
  - components, 7-2
  - copying, 7-9, 47-6
  - creating, 3-12, 7-9
    - for currency conversions, 43-7
    - for Personal Essbase, 39-3
    - prerequisites, 7-9
    - required privilege, 17-35
    - with Application Manager, 7-10
  - currency conversions and, 43-2
  - customizing, 1-5, 45-13
  - data distribution characteristics, 4-2
  - deleting, 17-35, 47-8
  - designing partitioned, 6-9
    - deciding which type, 6-10
    - scenarios for, 6-33, 6-36, 6-38
  - designing simple, 3-12, 3-13, 3-16
  - designing single-server, 5-1, 5-4
  - developing, 1-2, 3-2, 3-8
    - process summarized, 5-2

- applications (*Continued*)
  - file types listed, 7-2
  - implementing global security, 17-29, 17-31
  - inaccessible, 17-33
  - interruptions, 46-26
  - loading
    - with related database, 45-8
  - logging errors, 46-26
  - maintaining, 2-4
  - monitoring, 45-16, 46-41
  - naming rule, 7-12
  - not stopping, 45-6
  - opening, 7-7, 7-8
  - operations not supported, 3-2
  - overview, 7-2
  - partitioned
    - benefits of, 6-1, 6-2
    - choosing when not to use, 6-9
    - choosing when to use, 6-8
  - porting, 47-15, 47-18
    - creating backups for, 48-3
    - redefining information for, 47-19
    - to UNIX platforms, 47-16
  - renaming, 47-7
  - sample, xliv, 43-1
  - selecting, 7-7
  - size considerations, 15-4
  - starting, 7-3, 7-5, 45-4
    - from Server Agent, 45-5, 45-6
  - stopping, 7-5, 45-4, 45-5, 45-6
    - all opened, 45-12
    - archiving and, 48-2
    - with related database, 45-9
  - storing, 7-2
  - viewing, 21-2
    - information about, 46-2, 46-8, 46-28, 46-31
    - log file contents, 46-28
- Applications list box, 7-7
- Applications page (Server Information), 46-8
- Apply Outline Changes dialog box, 16-52
- applying
  - See also* setting
  - changes to outlines, 16-50, 16-52
  - locks, 17-37, 33-22, 33-23, 42-5, 46-40
  - privileges to groups, 17-14
  - skip properties, 9-6
- APPLYOTLCHANGEFILE command, 16-56
- ARBORDUMPPATH environment variable, 46-34
- ARBORPATH, 41-21
- .ARC files, 47-1
- architecture, 4-1
  - client-server features, 1-4
- archive files, 47-1
  - restoring from, 48-5
- archiving
  - data, 48-2
  - data targets, 6-18
- Area Definition dialog box
  - defining linked areas, 16-33, 16-34
  - defining replicated areas, 16-9
  - defining transparent areas, 16-22, 16-23
- Area Specific Member Mapping dialog box, 16-46
- areas
  - See also* partitions
  - changing shared, 16-48
  - defined, 6-6
  - Dynamic Time Series members in, 30-16
  - linked partitions, 16-33
  - mapping to specific, 16-42, 16-44
    - in Partition Wizard, 16-46
  - replicated partitions, 16-8
  - transparent partitions, 16-22
- Areas page (Partition Wizard), 16-8, 16-22, 16-33
- arithmetic operations
  - currency conversions, 43-25
  - formulas and, 5-31, 26-3
  - missing values and, 33-35
  - performing on fields, 22-22
  - performing on members, 9-18
  - prerequisite for, 22-23
  - report scripts, 36-21
  - specifying in data source, 14-25
- arranging
  - cells in blocks, 4-6
  - data blocks, 28-3
  - dimension build
    - position of fields in rules file, 14-17
  - dimensions in outlines, 28-6
  - fields, 22-5, 22-11
  - members in dense dimensions, 4-6
  - members in outlines, 4-6, 14-10

- ARRAY command, 31-12
  - usage example, 32-8
- arrays, 4-6
  - as variables, 31-12, 32-18
  - declaring, 32-8
- ASC command, 36-47
  - usage example, 37-55
- ascending sort order, 36-44
  - applying to output, 36-47
  - members in outlines, 8-21
- ASCII characters ignored in data loads, 20-7
- ASCII files
  - calc scripts in, 31-24
  - creating, 39-1, 39-7
  - cross-platform compatibility, 47-16
  - data sources, 20-8
  - dumping security information to, 45-5, 45-12
  - exporting outlines to, 8-11
  - loading, 23-2, 23-5
    - multiple, 23-7
  - opening, 21-5
- ASCII text, 26-11, 31-10
- assets, 27-4
- Assign Filters dialog box, 18-15
- assigning
  - See also* defining; setting
  - access levels to linked objects, 12-4
  - aliases to members, 11-1, 11-2, 11-3, 11-4
    - based on combinations, 11-5
  - attributes to dimensions, 43-14
  - filters to users and groups, 18-15
  - generic passwords, 17-10
  - privileges
    - global access, 17-30, 17-31, 17-34, 17-35
    - to users and groups, 17-5
  - properties to dimensions, 9-2, 9-3, 9-10, 9-11
    - overview, 9-2, 9-4, 9-6
  - properties to members, 5-19, 5-21
  - values to member combinations, 26-46
  - values to variables, 26-48
    - variance reporting properties, 9-9
- Associate Client Outline dialog box, 31-43
- Associate Server Outline dialog box, 21-22, 31-43
- associating
  - applications with calc scripts, 31-19, 31-25, 31-28
- attributes
  - automatically, 13-31
  - example rules file, 13-20
  - in Outline Editor, 10-28
  - through dimension build, 13-19
- calc scripts with outlines, 31-44
- databases with calc scripts, 31-19, 31-25, 31-28, 31-43
  - databases with reports, 35-26
  - filters with outlines, 18-4
  - members with report scripts, 36-4
  - multilevel attribute dimensions, 13-22
  - parents with members, 14-10
  - rules files
    - with data sources, 23-9
    - with outlines, 14-6, 14-9, 21-22
- asterisks (\*)
  - as codes in data source, 14-25
  - in application and database names, 7-12
  - in names in scripts and formulas, 8-16, 36-3
  - used as wildcard, 36-38
- ASYM command
  - entering in report scripts, 36-8
  - usage, 36-4
- asymmetric columns
  - changing headings, 36-8
  - creating, 37-39
  - dynamically calculating values, 29-18
  - in reports, 37-42
  - in source data, 20-22, 20-23
  - overriding groupings, 36-8
- asymmetric reports, 36-18
  - creating, 36-7, 37-40
  - defined, 36-7
  - formatting, 36-4
  - symmetric reports vs., 38-6
- at signs (@)
  - in dimension and member names, 8-15
  - in names in scripts and formulas, 8-16, 36-3
- attaching to databases. *See* connections

- attachments
  - See also* linked reporting objects
  - limiting size, 15-13
  - removing, 12-6
  - saving, 12-3
  - viewing in Application Manager, 12-6
- @ATTRIBUTE function, 10-37, 26-38
- attribute associations
  - allowing changes, 14-11
  - base dimensions, 10-5, 10-6
  - field type, 13-20
  - lost in cut or copy and paste, 10-28
  - requirements, 10-5
- Attribute Calculations dimension
  - accessing members from, 10-35
  - changing member names, 10-19
  - default member names, 10-20
  - described, 10-12
  - instead of consolidation symbols, 10-31
  - instead of member formulas, 10-31
  - members, default, 10-32
  - properties of, 10-31
  - retrieving multiple members from, 10-34
- ATTRIBUTE command, 6-28, 36-35
  - usage, 36-27
- attribute dimensions
  - associating with base dimensions in Outline Editor, 10-26
  - comparison with standard dimensions, 10-7
  - creating in Outline Editor, 9-13
  - defining in dimension build, 14-4
  - deleting members, 14-19
  - described, 5-20, 10-4
  - generation or level reference numbers, 14-16
  - members
    - overview, 10-5
    - prefixes and suffixes, 10-16
    - preventing creation, 13-20, 13-31, 14-11
  - multilevel
    - building and associating, 13-22
  - names as field types, 14-15, 14-24
  - outline design, 5-15
  - restructuring, 40-21
  - summary of dimension building rules, 13-30
- attribute dimensions (*Continued*)
  - tagging, 9-13
  - type, 9-14
  - using to avoid redundancy, 5-15
- attribute fields
  - defining using rules files, 13-18
  - position in rules file, 14-17
- attribute members
  - options in dimension build, 14-11
  - using in report scripts, 36-5
- attribute parent field type
  - example, 13-23
  - in header records, 14-24
  - in rules files, 14-15
  - nulls and, 13-7, 13-10
- attributes
  - advantages, 10-2
  - as values resulting from formulas, 10-13
  - associating
    - automatically, 13-31
    - in Outline Editor, 10-28
    - through dimension build, 13-19
  - associating aliases through dimension build, 13-26
  - Boolean type
    - changing default member names, 10-21
    - described, 10-7
    - duplicate values, 10-16
  - calculating
    - accessing calculated data, 10-35
    - Attribute Calculations dimension, 10-31
    - default calculation, 10-33
    - Essbase functions, 10-37
    - examples, 10-34
    - multiple calculations, 10-34
    - performance considerations, 10-35
    - process, 10-31
    - using attributes in formulas, 10-36
  - calculation functions, 26-38
  - date type, 10-7
    - changing the member name format, 10-22
    - duplicate values, 10-16
  - defined, 10-2
  - design considerations, 10-12

attributes (*Continued*)

- mapping, 16-28
- member name format, 10-17
- member names, 10-15
- numeric type
  - defined, 10-6
  - defining ranges, 10-23
  - duplicate values, 10-16
  - ranges, 10-7
  - size of ranges, 14-18
- process for defining in Outline Editor, 10-25
- setting types, 9-14
- shared member design approach, 10-13
- standard dimension design approach, 10-13
- text type, 10-6
- time-dependent values, 10-14
- types, 10-6
- UDAs
  - alternative design approach, 10-13
  - feature comparison, 10-9
- user-defined, 9-25
  - See also* UDAs
  - using in partitions, 6-28
- @ATTRIBUTEVAL function, 10-37, 26-44, 27-6
- ATTRPARENT. *See* attribute parent field type
- ATTRPROD.RUL, 13-20
- .ATX files, 47-1
- audio clips, 12-1
- audit log files, 23-2, 48-8
- Autoconfigure Dense/Sparse option, 14-21
- automating routine operations, 2-4
- average time balance property, 9-6
- average time balances specified in data sources, 14-25
- averages
  - Attribute Calculations dimension, 10-33
  - determining with formulas, 26-38, 27-3
  - for time periods, 30-1, 30-4
  - reporting examples, 37-32, 37-47
  - setting time balances, 9-6
- @AVG function, 26-38
- Avg member, Attribute Calculations dimension, 10-33
- @AVGRANGE function, 26-43, 27-3

## B

- B, time balance codes in data source, 14-25
- background startup, 45-2
- backing up databases, 48-1
- backslashes (\)
  - in application and database names, 7-12
  - in dimension and member names, 8-15
  - in formulas, 14-11
  - in names in scripts and formulas, 8-16
- backups
  - creating archive as, 48-2
  - creating by exporting, 48-3
  - restoring after system failures, 48-5
  - security, 17-2
- .BAK file, 17-2
- .BAK files, 47-4
- balance sheet accounts, 43-9
- bang command (!)
  - adding to report scripts, 36-2
  - terminating reports, 35-9
- .BAS files, 47-5
- base dimensions
  - associating with attribute dimensions in Outline Editor, 10-26
  - defined, 6-2, 10-4
  - member associations, 10-5
  - members
    - associating with attributes in Outline Editor, 10-28
    - attribute formulas, 10-36
    - attributes, 10-6
- Basic databases. *See* Demo Basic database; Sample Basic database
- BASIC directory, 37-1
- basic equations, 26-28
- basic reporting techniques, 35-10
  - creating simple reports, 35-2, 35-11
  - developing free-form reports, 35-28
  - editing report scripts, 35-11, 35-17, 35-18, 35-20
  - implementing security, 35-10
  - opening
    - report scripts, 35-14
  - opening report scripts, 35-16
  - running report scripts, 35-21, 35-26, 35-27
  - saving report scripts, 35-13

- .BAT files, 44-9
- batch files, 44-9
  - and ESSCMD scripts, 44-15
  - report scripts and, 44-15
  - running, 44-10
- batch mode, 41-7
  - command-line syntax for, 44-2
  - creating scripts for, 44-10
  - dimension building and, 14-26
  - dynamic calculations and, 29-4
  - error-handling, 44-16
  - examples
    - importing and calculating, 44-13
    - printing reports, 44-15
    - updating SQL scripts, 44-14
  - file-name extensions and, 44-4
  - overview, 44-9
  - shutting down server, 45-4
  - syntax, 44-2
  - updating outlines, 23-12
  - when to use, 44-1
- batch processing time, 6-22
- BEFORE command, 36-26
  - usage example, 37-36
- BEGINARCHIVE command, 48-2
  - partitioned applications and, 6-19
- BIN directory
  - ESSCMD files, 44-6
  - files stored in, 47-4
- bitmap compression, 41-27
  - described, 40-8
  - estimating block size, 15-5
  - overview, 40-11
  - specifying, 41-27
- bitmap dimensions, 33-16
- bitmap files, 47-4
- bitmap, calculator cache, 33-15, 33-17
- bitmaps, 12-1
- black squares (Report Viewer), 36-15
- blank fields
  - adding values to, 22-15
  - in data source, 20-16
  - in rules file, 20-17
- blanks. *See* white space
- block density (defined), 46-13
- block offsets, 24-9
- BLOCKHEADERS command, 36-8, 36-18
- blocks. *See* data blocks
- .BND files, 47-4
- Boolean
  - attribute dimension type, 9-14, 10-6
  - attributes
    - changing default member names, 10-21
    - described, 10-7
    - duplicate values, 10-16
  - expressions in rules files
    - for select/reject criteria, 21-16, 21-18
  - functions, in formulas, 26-4, 26-29
- Boolean expressions, 36-33
  - for select/reject criteria, 21-18
  - grouping, 37-58
  - usage example, 37-58
- Boolean operators, 36-32, 36-37
- BOTTOM command
  - entering in report scripts, 36-46, 36-48, 36-51
  - precedence, 36-46
  - restrictions, 36-50, 36-51
  - upper limits, 36-50
  - usage, 36-45
  - usage example, 37-52
- bottom-up
  - partitioning, 6-33
- bottom-up calculation, 33-12, 33-13
  - transparent partitions, 6-21
- bottom-up ordering
  - calculations, 9-15
  - dynamic builds, 13-3, 13-8, 13-10
  - examples, 13-9, 13-43
- bottom-up partitioning
  - defined, 6-2
- braces ( { } )
  - in dimension and member names, 8-15
  - in names in scripts and formulas, 8-16, 36-3
  - in report scripts, 35-9, 36-2
- brackets ( [ ] )
  - in application and database names, 7-12
  - in names in scripts and formulas, 8-16, 36-3

- BRACKETS command, 36-23, 36-26
  - overriding, 36-14
- branches
  - Calc Script Editor
    - aliases, 31-48
    - dimensions, 31-47
    - members, 31-45
  - data hierarchies, 3-6
  - Formula Editor
    - aliases, 26-25
    - dimensions, 26-24
    - members, 26-21
  - outlines, 8-3
  - sharing members, 13-40
- browser for editing objects, 12-2, 12-7
- budgets
  - allocating values, 32-9, 32-13
  - comparing actual to budgeted, 9-9, 26-8
  - example database for forecasting, 5-3
  - example for increasing, 31-6
  - generating and loading new, 32-5
  - getting variance, 26-39, 32-2
  - partitioned applications and, 6-10
  - variance reporting and, 5-31
- buffers, 15-2, 15-14
  - See also* caches
  - extracted data row cells, 38-2
  - increasing size, 29-12
  - setting internal, 38-2
  - sorted data, 38-4
- build methods
  - adding members
    - as children of specified parent, 13-16
    - as sibling with matching strings, 13-13
    - as siblings of lowest level, 13-15
  - bottom-up ordering, 13-8
  - creating
    - multiple roll-ups, 13-43, 13-44
  - creating shared members
    - at different generations, 13-38, 13-39
    - at same generation, 13-33, 13-35, 13-36
    - including branches, 13-40, 13-42
  - defined, 13-1
  - defining parent/child relationship, 13-11
  - description, 13-3
- build methods (*Continued*)
  - null processing, 13-7, 13-10
  - selecting, 13-4, 14-9
  - supported for associating attributes, 13-19
  - top-down ordering, 13-5
  - valid field types, 14-15
- BUILDDIM command, 14-2, 14-26
- building
  - See also* adding; creating
  - applications, 7-9
  - calc scripts, 31-3, 31-10, 31-36
    - in Calc Script Editor, 31-19, 31-39, 31-40
    - restrictions, 31-12
    - syntax for, 31-15
    - with Application Manager, 7-8
  - databases, 3-3, 3-4, 7-9
    - development process for, 5-4
    - example for, 5-3
    - prerequisites for, 5-4
  - dimensions, 14-26, 23-8, 23-10, 23-12, 23-13
  - dynamically. *See* dynamic builds
  - example for, 3-18
  - guidelines for, 5-7, 5-12
  - prerequisites, 21-15, 21-16, 23-2
  - restructuring databases and, 40-14
  - with data sources, 13-3, 13-5, 13-8, 13-11, 13-12
    - with dynamically calculated members, 29-23
    - with rules files, 13-2, 14-1, 14-2, 14-26
  - dynamic calculations, 29-8, 29-9
    - restrictions, 29-5, 29-10, 29-20
  - formulas, 26-15, 26-20
    - with Formula Editor, 26-13, 26-15
  - multiple roll-ups, 13-44
  - reports, 36-1, 36-7, 38-6
    - basic techniques, 35-2, 35-11
    - free-form, 35-28
    - with API function calls, 36-52
  - shared members dynamically, 13-32, 13-33, 13-37, 13-40
- builds. *See* dynamic builds
- business models
  - checklist for creating, 5-10
  - creating as a part of database design, 5-6

**C**

## Cache Memory Locking

- enabling with Application Manager, 41-13
- enabling with ESSCMD, 41-13
- updating, 41-13

## caches

- as storage units, 15-2
- calculator, 33-3, 33-15, 33-22
- determining size, 15-14
- disabling, 33-21
- getting information about, 46-11, 46-14
- locking memory for, 41-12
- managing, 40-3, 40-5
- optimizing read/writes, 24-10, 24-11
- setting size, 15-16, 33-14, 33-18
  - with Application Manager, 41-8, 41-10
  - with ESSCMD, 41-9, 41-10
  - with Hyperion Essbase kernel, 41-8, 41-9
- updating, 41-9, 41-10, 41-12

## CALC ALL command, 31-11

- block ordering and, 28-14
- currency conversions, 43-26
- dynamic calculations and, 29-2
- dynamically calculated members and, 29-8
- Intelligent Calculation and, 33-32, 33-33, 34-5, 34-14
- partitioned applications and, 6-19
- usage overview, 25-7, 28-3, 32-3

## CALC AVERAGE command, 31-11

CALC COL command. *See* CALCULATE COLUMN command

## CALC command, 25-11

## CALC DIM command, 31-11

- dynamically calculated members and, 29-8
- Intelligent Calculation and, 34-3, 34-6, 34-14, 34-15, 34-16
- usage examples, 31-51, 32-5, 32-6

## CALC FIRST command, 31-11

## CALC LAST command, 31-11

CALC ROW command. *See* CALCULATE ROW command

## Calc Script button (Application Manager), 7-8

## calc script calculations, 25-3

## Calc Script Editor

- building calc scripts, 31-19, 31-39, 31-40
  - changing scripts, 31-20
  - checking syntax, 31-5, 31-48, 31-49
  - copying scripts, 31-29
  - expanding/contracting dimensions, 31-47
  - expanding/contracting member branches, 31-45
  - finding and replacing text, 31-42
  - opening, 31-18
  - printing calc scripts, 31-35
  - saving calc scripts, 31-24, 31-29
  - searching for members, 31-46
  - undoing last action, 31-36
- calc script files, 31-20, 31-24
- naming, 31-26, 31-28
  - opening, 31-32, 31-34
  - selecting, 31-21, 31-22
- calc scripts, 5-31
- adding
    - aliases, 31-48
    - comments, 31-15
    - equations, 26-28, 31-37
    - formulas, 31-16, 31-36, 31-38, 31-51
    - text, 31-39
  - aggregating missing values in, 33-36
  - applying conditions, 26-29
  - attaching to databases, 31-43
  - building in Calc Script Editor, 31-10, 31-19, 31-39, 31-40
  - calculation order, 28-24
  - changing, 31-20, 31-21, 31-22
  - checking syntax, 31-5
  - clearing databases after export, 48-4
  - copying, 31-29, 47-12
  - creating with Application Manager, 7-8
  - currency conversions, 43-26, 43-27
  - declaring variables in, 31-12, 31-52
  - defined, 7-4, 31-2
  - defining
    - as default, 25-7
    - execution access, 17-26
  - deleting, 31-35, 31-36
  - displaying completion notices with, 33-5
  - dynamic calculations and, 29-8, 29-21

- calc scripts (*Continued*)
  - examples, 31-3, 31-14, 31-55, 32-1
  - formulas in, 26-5
  - generating statistical information, 33-4
  - grouping formulas and dimensions, 31-50, 31-51, 33-13
  - inserting
    - calculation commands, 31-11, 31-12, 31-13
    - variables, 26-48
  - Intelligent Calculation and, 31-50, 31-54, 34-2, 34-6
  - loading, 31-20, 31-22, 31-23
  - names with special characters, 8-16, 8-17
  - overriding default calculation order, 28-4
  - performing multiple calculation passes, 34-3
    - examples, 34-14, 34-15, 34-16
  - printing, 31-35
  - restrictions, 31-12
  - running, 31-31
    - aborted transactions and, 42-12
    - calculating Sample Basic, 31-7
    - information messages after, 31-50
    - on partitioned applications, 31-57
    - with Application Manager, 7-8
    - with ESSCMD, 31-24
  - saving, 31-5, 31-24, 31-25
    - on client workstations, 31-27, 31-29
  - syntax for
    - checking, 31-5, 31-48
    - guidelines, 31-15
  - troubleshooting, 31-48
  - two-pass calculations, 33-26, 33-31, 33-32, 33-33, 33-34
  - UDAs and, 9-25
  - undoing changes, 31-36
  - unlocking, 46-40
  - usage overview, 31-1, 31-2, 31-50
  - viewing log file contents, 31-9
- CALC TWOPASS command, 31-11
  - usage examples, 33-33, 34-14
- CALCDEFAULT command, 25-11
- CALCHASHTBLMEMORY setting, 33-4
- CALCLINE command, 25-11
- CALCLOCKBLOCK setting, 33-23
- CALCOPTCALCHASHTBL setting, 33-4
- CALCOPTFRMLBOTTOMUP setting, 33-8
- CALCULATE COLUMN command, 36-16
  - usage examples, 37-33, 37-41
- Calculate Database dialog box, 25-10
  - canceling operations, 25-11
  - running calc scripts, 31-34
- calculate privilege, 25-12
  - defined, 17-26, 17-30
  - filters and, 18-2
- CALCULATE ROW command, 36-20, 36-21
  - restrictions, 36-51
  - usage examples, 37-44, 37-46, 37-48
- calculated columns
  - adding totals, 36-20
  - clearing values, 36-19, 36-20
  - creating, 36-16, 37-32, 37-40
- calculated data, 5-25
  - filtering, 18-2
  - formatting, 36-16, 36-20
- calculated values vs. input values, 25-1
- calculation commands, 31-2, 31-10
  - computation, 31-11
  - control, 31-12
  - declaring temporary variables for, 31-12
  - iterating through, 31-12
  - specifying global settings, 31-13
- calculation scripts. *See* calc scripts
- calculations, 4-6, 13-46, 28-23
  - See also* calc scripts; dynamic calculations
  - aborted transactions and, 42-12
  - account reporting values, 30-1
  - across multiple processors, 6-8
  - ad hoc, 36-16
  - adding formulas to, 5-31, 26-1, 26-5, 26-6
    - with Formula Editor, 26-13, 26-15
  - assigning constants, 33-8
  - associating with specific database, 31-19, 31-25, 31-28, 31-43
  - attributes
    - accessing calculated data, 10-35
    - Attribute Calculations dimension, 10-31
    - default calculation, 10-33
    - described, 10-30

calculations (*Continued*)

- examples, 10-34
- multiple calculations, 10-34
- performance considerations, 10-35
- process, 10-31
- using attributes in formulas, 10-36
- basic concepts, 25-4
- blocks with dirty status, 34-14
- bottom-up, 33-12, 33-13
- calc scripts vs., 31-2
- caution for changing outlines and, 8-21, 8-22
- checklist for defining, 5-36
- concurrent and memory usage, 33-14
- controlling flow, 26-30, 31-2, 31-12
- currency conversions, 43-26, 43-27, 43-30
  - links and, 43-24
- dates, 5-30
- default, 33-27
  - overriding, 31-2
  - setting, 25-7, 34-3, 34-5
- defining global behavior, 31-13
- defining order, 28-1
  - cells, 28-14, 28-22
  - data blocks, 28-11
  - dimensions, 28-6
  - forward references and, 28-8
  - members, 28-4, 28-5
- designing for optimal, 33-2
- difference between actual and budgeted, 9-9
- displaying completion notices, 33-5
- displaying settings, 33-4
- entire database, 31-11
- extending capabilities, 5-32
- failing, 17-37
- first-time, 34-6
- fixing unexpected results, 22-23, 23-17
- getting current state, 25-10
- getting information about, 46-17
- handling missing and zero values, 9-7
- members across multiple parents, 9-23
- members in outlines, 31-11
- members with different operators, 9-17
- missing values and, 31-13, 31-53, 33-35
- monitoring, 33-4
- monthly assets, 27-4

calculations (*Continued*)

- multiple databases, 31-58
- operator precedence, 9-17
- optimizing, 33-1, 33-21
  - with attributes, 10-35
  - with bottom-up calculation, 33-12
  - with calc scripts, 31-50
  - with calculator cache, 33-15
  - with dynamic calculations, 29-5, 29-13
  - with Intelligent Calculation, 34-1
  - with interdependent values, 26-33
  - with two-pass calculations, 33-24
- overview, 25-1, 25-8, 25-12
- partial list of members, 31-55
- partitioned applications
  - with replicated partitions, 6-13
  - with transparent partitions, 6-18, 6-19, 6-21, 6-22
- performance and multi-users, 33-23
- performance, with attributes, 10-35
- performing multiple passes, 34-3, 34-4, 34-8, 34-13
  - examples, 34-14, 34-15, 34-16
- period-to-date values, 27-1, 30-7
- preventing delays, 33-24
- recovering, 48-7
- relationships between members, 5-31
- report scripts, 36-16, 36-20
  - examples, 37-32, 37-40, 37-44
- rolling averages, 27-3
- running, 25-11
  - default, 28-14
  - in batch mode, 29-4, 44-13, 44-14
- setting up two-pass, 5-34, 9-15, 14-25
- shared members, 28-26
- single-server applications, 5-25, 5-26, 5-27, 5-28, 5-29, 5-31
- specified dimensions, 31-11
- statistics, 26-41
- stopping, 25-11
- subsets of data, 31-12, 31-54, 31-55
  - example, 32-3
  - with Intelligent Calculation, 34-4
- testing, 33-2

- calculations (*Continued*)
  - top-down, 33-12, 33-13
  - types described, 25-2, 25-3
  - variance performance, 5-31
  - with a series of dimensions, 31-51
  - with a series of member formulas, 31-51
  - year-to-date values, 27-3
- calculator cache
  - as storage unit, 15-2
  - bitmap, 33-15, 33-17
  - disabling, 33-21
  - maximum size, 33-15
  - optimizing, 33-3, 33-22
  - overview, 33-15
  - setting size, 33-14, 33-18, 33-21
- calculator hash tables, 33-4, 33-22
- canceling
  - archiving operations, 48-3
  - calculations, 25-11
  - changes to calc scripts, 31-36
  - ESSCMD operations, 44-3, 44-8
- captures, 36-20, 46-33
- carriage return as file delimiter, 20-6
- case conversions, 20-11, 22-16
- case sensitivity
  - field type designation in header record, 14-24
  - member names, 8-24
- case-sensitive names
  - converting case, 20-11, 22-16
  - defining selection criteria for, 21-16, 21-17
  - report scripts, 36-2
  - setting for database, 8-15, 8-24
- case-sensitive passwords, 17-13
- case-sensitive searches
  - Calc Script Editor, 31-42
  - Field Properties dialog, 22-15
  - Formula Editor, 26-19, 26-23
  - Report Editor, 35-17
- case-sensitivity in ESSCMD, 44-5
- cash flow, 26-33, 26-45
- catalogs, 12-3, 15-13, 40-5
- category conversion property, 9-10
- CCONV command, 29-20, 31-11
  - usage examples, 43-26, 43-27
  - usage overview, 34-20, 43-26
- CCONV TOLOCALRATE command, 43-26
- CCTRACK setting, 43-30
- cells, 3-8
  - accessing, 4-6
    - simultaneously in different databases, 6-23
  - annotating, 12-2
  - copying range of, 31-54
  - determining calculation order for, 28-14, 28-22
    - examples, 28-15, 28-16, 28-18, 28-20
  - empty, 4-7
    - caution for storing, 4-16
  - linking objects to, 12-1
  - mapping to targets, 6-3
  - ordering in blocks, 4-6
  - partitioning and, 6-12, 6-17, 6-23
  - removing linked objects, 12-6
  - returning unique values for, 4-5
  - shaded, 3-9
- centering data, 36-9, 36-17
  - example, 37-27
- centralized data repositories, 5-3
- .CFG files, 38-5, 47-4
  - See also* configurations
- change log files, 16-48, 40-19
  - contents described, 46-37
  - enabling/disabling, 46-38
  - format example, 46-36
  - overview, 46-35
  - restructuring and, 40-20
  - setting size, 46-39
  - updating, 16-53
- Change Password dialog box, 17-13
- changes
  - affecting only outlines, 40-15
  - applying to outlines, 16-50, 16-52
  - discarding partition, 16-41
  - impacting restructures, 40-21, 40-27
  - overriding incremental restructuring, 40-19
  - overwritten, 6-11
  - relational storage manager and, 40-20
  - tracking outline, 16-48
  - unable to make, 16-5
  - undoing, 31-36
  - viewing outline, 46-35

## changing

- See also* editing; altering
- access privileges, 17-22, 17-27, 17-28, 17-31
- alias table names, 11-9
- aliases, 14-10
- attribute associations, 14-11
- calc scripts, 31-20, 31-21, 31-22
- consolidations, 5-21
- data, 6-11, 22-22, 26-48
  - prerequisite for, 17-37
- data values, 22-22
- default storage properties, 5-21
- dimension names, 8-10
- dimension properties, 13-4, 14-6, 14-10, 14-23
- dimensions, 14-10, 40-27
  - default configurations, 8-9
  - for currency conversions, 43-8
- formulas, 14-11, 26-13
- headings in reports, 36-8
- Hyperion Essbase kernel settings, 41-2, 41-3
  - scope and precedence, 41-4
  - with Application Manager, 41-5
  - with ESSCMD, 41-3, 41-7
- linked partitions, 12-9
- member combinations for linked objects, 12-9
- member names, 8-10
- members, 14-10
- outlines, 8-1, 16-47, 16-52, 40-15
  - caution for, 8-21, 8-22, 23-10
  - dynamically, 13-3
  - for currency conversions, 43-7
  - with rules files, 23-10
- passwords, 17-13
- report layouts, 36-26
- security settings, 17-12, 17-21
- system password, 45-5, 45-11

character searches. *See* searches

character strings. *See* strings

## characters

- calc scripts, 31-15
- end-of-file markers and special, 24-6
- forbidden at the beginning of a name, 8-15
- formulas, 26-6
- ignored in data sources, 20-7
- ignored in report extraction, 36-3

characters (*Continued*)

- maximum
  - in aliases, 11-4
  - in application and database names, 7-12
  - in application names, 7-10
  - in database names, 7-12
  - in dimension names, 8-15
  - in member names, 8-15, 11-4
  - in URLs, 12-5
- numeric in member names, 20-5
- quoting in scripts and formulas, 8-16, 8-17
- valid in numeric data fields, 20-4

## checking

- disk space, 5-2
- forward references, 28-9
- syntax
  - Calc Script Editor, 31-5, 31-48, 31-49
  - Formula Editor, 26-25, 26-26

## check-out facility, 47-14

.CHG files, 16-48, 47-2

- updating, 16-53

## child

- See also* parent/child relationships
- adding as member of specified parent, 13-16
- as only member (implied sharing), 9-24
- calculation order for outlines, 9-15
- checking for, 26-29
- consolidation properties and, 9-16
- currency conversions and, 43-10
- defined, 3-6
- dimension as, 8-18
- member as, 8-19, 8-22
- rolling up, 13-37
- shared member as, 9-23

## child field type

- in header records, 14-24
- in rules files, 14-15
- sharing members, 13-36, 13-39, 13-42

## child processes, 45-16

CHILD. *See* child field type

@CHILDREN function, 26-36

CHILDREN command, 36-44

- entering in report scripts, 39-14
- usage, 36-27

- choosing
  - alias tables, 26-25, 31-48
  - applications for loading, 7-7
    - with ESSCMD, 44-7
  - build methods, 13-4, 14-9
  - calc script files, 31-21, 31-22
  - data sources, 6-9, 16-4, 21-1, 23-3, 23-5
    - using Windows, 23-7
  - data targets, 16-4
  - data to partition, 6-3, 6-8, 6-9
  - databases for loading, 7-7
    - with ESSCMD, 44-7
  - dimension type, 4-8, 4-11, 4-13, 4-14
    - guidelines for, 5-7
  - fields, 22-2
  - members
    - for dynamic calculations, 29-13
  - members for column groupings, 36-7
  - members for dynamic calculations, 29-13, 29-15
  - members for report scripts, 36-27, 36-29, 36-39
    - from Dynamic Time Series, 36-31
    - with ATTRIBUTE command, 36-35
    - with Boolean operators, 36-32
    - with conditions, 37-57
    - with substitution variables, 36-33
    - with TODATE command, 36-36
    - with user-defined attributes, 36-37
    - with wildcards, 36-38
    - with WITHATTR command, 36-36
  - multiple dimensions and members, 37-36
  - multiple files, 23-7, 23-8
  - multiple members, 8-22
  - partition type, 6-10, 6-15, 6-20, 6-26, 6-27
    - in Partition Wizard, 16-3, 16-5, 16-21, 16-31
  - records, 21-15, 21-16, 21-18
  - report script files, 35-15, 35-16
  - SQL data sources, 23-4
  - values for dynamic calculations, 29-6, 29-15
    - guidelines for, 29-7, 29-8
- clean status
  - caution for sparse dimensions, 31-50
  - clearing data and, 34-19
  - Intelligent Calculation and, 34-4
  - marking blocks with, 34-2, 34-6
  - partial calculations and, 31-54
- Clear Alias Table dialog box, 11-11
- Clear Data Combinations page (Data Load Settings), 22-24
- CLEARALLROWCALC command, 36-20
- CLEARBLOCK command, 31-53, 34-19
- CLEARBLOCK DYNAMIC command, 29-4, 29-20, 31-53
- CLEARDATA command, 29-20
  - partitioned applications and, 6-18
  - usage overview, 31-53, 34-19
- clearing
  - See also* deleting
  - alias tables, 11-11
  - data, 6-18, 22-23, 31-53, 34-19
    - caution, 12-3
    - in calculated columns, 36-19, 36-20
  - locks, 17-37
  - log file contents, 46-29, 46-32
  - member combinations, 22-25
  - values in transparent partitions, 22-24
- CLEARLOGFILE setting, 46-29, 46-32
- CLEARROWCALC command, 36-20
- CLEARUPDATESTATUS command, 32-3
- client
  - interfaces
    - accessing linked objects and, 12-4
  - workstations
    - as used in Hyperion Essbase, 7-2
    - caution for loading data from, 24-11
    - troubleshooting connections, 24-4
- CLIENT directory
  - location on client for applications, 8-12
- client interfaces, 1-5
- client window, 7-7
- client workstations
  - caution for improper shutdown, 17-33
  - deleting calc scripts, 31-36
  - recovering from improper shutdowns, 17-33
  - writing calc scripts to, 31-27, 31-29
- clients, 1-4, 45-12, 45-15
  - communicating with server, 45-1
  - configuring, 1-5
  - types described, 1-5, 45-13
- client-server applications
  - See also* single-server applications
  - development guidelines, 3-2
  - overview, 1-4

- client-server model, 1-4, 45-12, 45-14
- Close dialog box
  - saving partition definitions, 16-41
- closing
  - See also* exiting; quitting; stopping
  - applications, 7-5, 45-4, 45-5, 45-6
    - archiving and, 48-2
    - multiple simultaneously, 45-12
    - with related database, 45-9
  - databases, 7-6, 45-8, 45-9
  - Log Viewer, 46-29
  - Partition Wizard, 16-41
- .CNT files, 47-4
- codes in data source
  - setting member properties, 14-24
- COLCALC1.REP, 37-34
- COLCALC2.REP, 37-41
- COLCALC3.REP, 37-43
- COLGROUP.REP, 37-8
- COLHEADING command, 36-6
- Collapse to Ancestor command, 8-4
- collapsing
  - alias member lists, 11-6
  - Calc Script Editor lists, 31-45
  - Formula Editor lists, 26-21
  - outline hierarchies, 8-3
- colons (:)
  - in application and database names, 7-12
  - in formulas, 26-35
  - in names in scripts and formulas, 8-16, 36-3
- column calculation commands, 36-16
- COLUMN command
  - entering in report scripts, 36-4
- column formatting commands, 36-10, 36-13, 36-14, 36-15
- column headings
  - adding to calculated columns, 36-17
  - adding to reports, 35-29, 36-4
  - changing, 36-8
  - defined, 35-7
  - displaying, 36-6
- column headings
  - (Continued)*
  - for loading asymmetric columns, 20-23
  - multi-line, 36-17
  - names truncated, 36-13
  - repeating, 36-18
  - SQL data sources, 21-8
  - suppressing, 36-14
- column widths, 36-9
- columns
  - See also* fields
  - adjusting length, 36-13
  - calculating in reports, 37-32, 37-40
  - creating asymmetric, 37-39
  - creating calculated, 36-16, 36-20
  - defining as fields, 22-21
  - fixed-width, 21-11
  - formatting, 20-22
  - formatting for data load, 20-22, 20-23
  - mapping specific fields only, 22-3
  - nesting in reports, 37-6
  - numbering, 36-18
  - ordering in data sources, 13-11
  - overriding grouping in reports, 36-8
  - parent/child data sources, 13-11
  - replacing empty fields with text, 22-15
  - restricting number of, 37-22
  - setting as data field, 20-16
  - switching with rows, 37-8
  - symmetry in reports, 36-7
  - using attributes in, 36-5
- combining fields. *See* joining fields
- command prompt, 44-6
- command syntax, xliii
- command-line interface
  - See also* ESSCMD; server console
  - ESSCMD, 44-1
  - running batch files, 44-10
- commands, 31-2
  - See also* specific command
  - calculation types listed, 31-10
  - case sensitivity, 44-5
  - caution for processing, 44-8
  - computation, 31-11
  - control, 31-12

- commands (*Continued*)
  - data extraction, 35-9, 36-1, 36-27
  - declaring temporary variables for, 31-12
  - displaying help for, 45-5
  - entering in ESSCMD scripts, 44-8
  - entering in report scripts, 36-2
  - ESSCMD syntax, 44-2
  - global calculation settings, 31-13
  - iterating through, 31-12
  - member selection, 36-27, 36-44
  - page layout, 36-3, 36-6
  - performance-related, A-2
  - report formatting, 35-9, 35-29, 36-1, 36-2, 36-8
    - caution for usage, 36-48, 36-51
  - report output, 35-9
  - Server Agent, 45-4
  - sorting, 36-44, 36-46
  - startup in server console, 45-2
- commas (,.)
  - as file delimiter, 20-6
  - displaying in reports, 36-26
  - in application and database names, 7-12
  - in data fields, 20-4
  - in dimension and member names, 8-15
  - in formulas, 26-35
  - in header records, 21-13
  - in member combinations, 22-25
  - in names in scripts and formulas, 8-16, 36-3
  - suppressing in reports, 36-14, 36-23
- COMMAS command, 36-23, 36-26
  - overriding, 36-14, 36-23
- comment fields, 22-2
- comments
  - See also* annotating
  - adding to dimensions, 9-27
  - adding to members, 9-27
  - calc scripts, 31-15
  - display options, 8-25
  - report scripts, 36-3
  - storing, 12-3
    - viewing in Application Manager, 12-8
- commission, 26-7, 26-31, 27-5
  - returning from calc scripts, 31-16
- Commit Block setting, 41-16, A-4
- Commit Row setting, 22-23, 41-16, A-4
- commits, 42-9
  - data loads failing and, 22-23, 24-5
  - initiating, 42-10
  - managing, 40-6
  - rollbacks and, 42-12
  - setting isolation level, 41-15, 41-17
    - with ESSCMD, 41-7, 41-18
  - updating isolation level, 41-19
- committed access
  - about, 42-2
  - caution for, 41-16
  - defined, 41-15
  - locks and, 40-6, 42-7, 42-9
  - rollbacks and, 42-11
  - setting, 41-16, 41-17, 41-18
- common currency, 43-1
- communications
  - client-server models, 45-1, 45-12, 45-14
- communications protocols, 45-14
- Company database
  - provided with Hyperion Essbase, xliv
- comparing
  - data values, 3-11, 9-9, 38-5
  - timestamps for outlines, 16-49
- compilers, 1-5, 45-13
- complete loads, 23-13
- completion notices, 33-5
- complex formulas, 29-7, 33-7, 33-12
- @COMPOUND function, 26-45
- compounded interest, 26-45
- @COMPOUNDGROWTH function, 26-45
- compression, 4-7
  - checking compression ratio, 40-11
  - enabling/disabling, 40-10
  - estimating block size, 15-5
  - fragmentation allowance, 15-10, 40-14
  - optimal configuration, 40-10
  - overview, 40-11
  - processes described, 40-8
  - ratio, 46-14
  - repetitive values, 40-8, 40-9, 40-10
  - specifying, 41-27, 41-28
  - types defined, 40-8
  - updating, 41-29
  - values (ESSCMD), 41-28
  - viewing settings information, 46-11

- computing data relationships, 3-4
- concatenating fields, 22-6
- concurrent calculations, 33-14, 34-8
  - overview, 34-12
- conditional blocks, 26-29
- conditional equations, 31-37
- conditional operators, 5-31, 26-3
- conditions
  - adding to report scripts, 36-32, 36-45
    - examples, 37-54, 37-55, 37-57
  - logical, 5-31, 26-29
  - setting for ESSCMD, 44-11
  - specifying
    - for data loads, 21-16, 21-17, 21-18
    - in formulas, 26-29, 26-31
  - testing, 26-3, 26-4
- configurable variables, 36-45
  - setting, 38-2
- configuration files, 38-5, 47-4
- configurations
  - activating change logs, 40-20
  - activating outline change log, 46-35, 46-38
  - changing, 14-21
  - checking, 46-1
  - clearing log files, 46-29, 46-32
  - clients, 1-5
  - controlling log file contents, 46-30
  - currency calculations, 43-30
  - database, 33-2
  - dimensions
    - automatic, 4-8, 8-9, 14-21
    - changing default setting, 8-9
    - determining optimal, 4-9, 4-13
  - enabling incremental restructuring, 40-19
  - Hyperion Essbase kernel settings and, 41-3
  - Intelligent Calculation, 34-5
  - logging spreadsheet updates, 48-8
  - resetting error log
    - files, 46-27
    - limits, 24-2
  - setting number of threads, 45-15
  - setting outline change file size, 46-39
  - viewing settings information, 46-3, 46-5
- Confirm Delete dialog box, 47-8, 47-11
- Confirm Password text box, 17-10, 17-17
- Confirmation dialog box, 8-27
- Connect page (Partition Wizard), 16-3
- connection passwords, 45-11
- connections
  - communications protocols, 45-14
  - databases with data cells, 6-23
  - losing, 16-62
  - multiple servers, 1-4
  - partitioned applications, 16-6, 16-21, 16-32
  - preventing system failures, 6-8, 6-12
  - terminating, 17-33
    - for specific user, 17-35, 17-38
  - troubleshooting, 24-4
- Connections dialog box, 17-38
- consecutive values, 20-19
- consistency. *See* data integrity
- console. *See* server console
- consolidation, 3-3, 3-7
  - changing, 5-21
  - codes in data source, 14-25
  - default order, 5-27
  - defined, 3-4
  - defining for members, 28-5, 28-7
  - display options, 8-25
  - excluding members from, 9-18
  - fixing unexpected results, 22-23, 23-17
  - operators listed, 9-17
  - performance considerations, 33-6
  - restrictions, 5-21
  - specifying requirements for, 5-17
- consolidation paths
  - defining, 5-26, 5-28
    - checklist for, 5-28
- consolidation properties
  - described, 9-16
  - setting, 9-18, 14-24
  - shared members and, 9-23
- consolidations (in Hyperion MBA). *See* defaults,
  - calculations
- constants, 33-8, 36-21
- contracting. *See* collapsing
- control information, 42-11
- conventions. *See* naming conventions

- conversions, 43-6
  - See also* currency conversions
  - case, 20-11, 22-16
  - positive/negative values, 22-27
  - white space
    - in data loads, 20-11
    - to underscores in data load, 22-19
- converting
  - dates, 26-46
  - spaces to underscores, 22-19
- coordinates (data values), 3-8
- Copy Alias Table dialog box, 11-9
- Copy Filter dialog box, 18-12
- Copy Group dialog box, 17-18
- Copy to Server Object dialog box, 47-12
- Copy User dialog box, 17-17
- COPYAPP command, 47-7
- COPYDB command, 8-4, 39-4, 47-9
- COPYFILTER command, 18-13
- copying
  - See also* duplicating; replicating
  - alias tables, 11-9
  - applications, 7-9, 47-6
  - base dimension members, 10-28
  - calc scripts, 31-29
  - data, 6-18, 29-20, 31-54, 34-3, 34-19
  - database outlines, 8-4
  - databases, 7-9, 47-9
  - filters, 18-12
  - from data targets, 6-10
  - objects, 47-12
  - outline files, 39-2, 39-4, 39-6
  - outlines
    - with Application Manager, 39-5
  - security profiles, 17-16, 17-18
  - text
    - in calc scripts, 31-41
    - in formulas, 26-18
    - in report scripts, 35-20
- @CORRELATION function, 26-41
- correlation coefficient, calculating, 26-41
- corrupted data, 40-8, 48-2
  - operations causing, 48-5
- cost of goods, 26-28
  - example for allocating, 32-7
- @COUNT function, 26-41
- Count member, Attribute Calculations dimension, 10-32
- count, calculating, 26-41
- country dimension
  - creating, 43-11
  - currency applications, 43-3
  - description, 5-20
  - specifying, 9-11
  - usage overview, 9-11
- country-specific data. *See* locales, xli
- .CPL files, 47-4
- crashes
  - getting information about, 46-27
  - minimizing, 6-8, 6-12
  - recovering from, 24-5, 48-5, 48-6
    - suggested procedures, 48-7
  - restructuring and, 40-17
  - transaction rollbacks and, 42-11
- Create Alias Table dialog box, 11-7
- Create Blocks on Equations option, 33-9
- Create Currency Database dialog box, 43-20
- Create Field Using Join dialog box, 22-7, 22-8
- Create Field Using Text dialog box, 22-10
- Create New Application dialog box, 7-10, 39-3
- Create New Database dialog box, 7-11, 39-3
- Create/Delete Applications privilege type, 17-7
- Create/Delete Users/Groups privilege type, 17-6
- CREATEAPP command, 7-10
- CREATEDB command, 7-12
- CREATEGROUP command, 17-16
- CREATELOCATION command, 7-21
- CREATEUSER command, 17-12
- CREATEVARIABLE command, 7-16
- creating
  - See also* adding; building
  - alias combinations, 11-5
  - alias tables, 11-3, 11-7
  - aliases, 11-3, 11-4
    - based on member combinations, 11-5
    - caution for, 16-6, 16-21, 16-32
    - guidelines for, 11-2

creating (*Continued*)

- applications, 3-12, 7-9
  - for currency conversions, 43-7
  - for Personal Essbase servers, 39-3
  - prerequisites, 7-9
  - required privilege for, 17-35
  - with Application Manager, 7-10
- backups, 48-2, 48-3
- data blocks, 34-10
- data load rules, 20-9
- database outlines, 3-3, 3-4, 4-8, 5-19
  - example for, 3-13
  - from data sources, 13-2
  - guidelines for, 5-14
  - prerequisites for, 5-4
  - with Outline Editor, 8-1, 8-2
- databases, 4-16, 5-16, 7-9
  - for Personal Essbase servers, 39-3
  - process summarized, 7-5
  - with Application Manager, 7-11
- Dynamic Calc And Store members, 9-21
- Dynamic Calc members, 9-21
- ESSCMD script files, 44-1, 44-10
- fields, 22-9, 22-10, 24-12
  - with joins, 22-6, 22-7
- filters, 18-3, 18-6
  - for partitioned databases, 6-30, 6-31
- formulas, 5-31, 27-1
  - examples, 26-8, 26-28, 26-31
  - syntax for, 26-6
  - with Formula Editor, 26-11, 26-13
  - writing equations, 26-27
- header records, 21-11, 21-13
- linked partitions, 6-25, 6-26
- linked reporting objects, 12-2
- member groups, 9-22, 24-8
- multiple roll-ups, 13-43, 13-44
- outline files, 39-7
- outlines
  - for currency conversions, 43-17
- outlines for currency conversions, 43-21
- partition definition files, 16-40
- partitions, 16-1, 16-3, 16-4, 16-20, 16-31
  - for currency conversions, 43-4, 43-26
  - process summarized, 6-7

creating (*Continued*)

- queries, 3-14
- replicated partitions, 6-11, 6-12, 6-13
- report files, 35-24
- report scripts, 36-1, 36-2
  - basic techniques, 35-2, 35-12
  - in ESSCMD, 35-5
  - parts described, 35-9
  - with Application Manager, 7-8
- rules files, 7-8, 13-2
  - process overview, 21-1
- scripts for batch processing, 44-10
- shared members, 9-23, 13-40, 13-42, 13-44
  - for different generations, 13-38, 13-39
  - for same generation, 13-33, 13-35, 13-36
  - guidelines for, 9-23, 13-4
  - with Outline Editor, 9-25
  - with rules files, 13-32
- shared roll-ups, 13-44
- substitution variables, 7-15
- text files, 39-1, 39-7
- trace files, 46-33, 46-35
- transparent partitions, 6-16
- two-dimensional reports, 39-13, 39-14
- UDAs, 9-25, 9-26
- user groups, 17-11, 17-14, 17-16
- cross-dimensional members
  - in formulas, 26-3, 26-29, 33-7, 33-12
- cross-dimensional operator (→), 4-7, 33-11
  - inserting in calc scripts, 31-40
  - inserting in formulas, 26-3, 26-16
  - overview, 26-46, 33-10
  - usage examples, 3-10, 26-47
- crossstab defined, 10-3
- .CSC files, 47-2, 31-20, 31-24
- cubes, 3-8, 25-5
- CURCAT. *See* currency category field type
- @CURGEN function, 26-44
- CURHEADING command, 36-51
- @CURLEV function, 26-44
- curly braces ( { } ). *See* braces ( { } )
- CURNAME. *See* currency name field type

- currency
    - ad hoc report, 43-5
    - defining country-specific, 9-11
    - effects of adding new markets, 43-2
    - exchange rates, 9-10, 43-2, 43-22
      - calculation options, 43-30
    - formatting in reports, 36-26
  - currency applications
    - creating, 43-7
    - overview, 43-2, 43-3, 43-5
    - sample, 43-1
  - currency category, 43-9
    - dimension (example), 43-5
  - currency category field type
    - in header records, 14-24
    - in rules files, 14-15
    - nulls and, 13-7, 13-10
  - CURRENCY command, 36-51
    - overriding, 36-14
    - usage overview, 43-29
  - Currency Conversion
    - description of product, 43-1
  - currency conversions, 31-11
    - adding members, 43-22, 43-25
    - base to local exchange, 9-12
    - calculating in report scripts, 36-51
    - creating outlines for, 43-17, 43-18, 43-21
    - creating partitions for, 43-4, 43-26
    - defining, 43-2
    - dynamic calculations and, 29-20
    - getting information about, 46-12
    - Intelligent Calculation and, 34-20
    - keeping local and converted values, 43-27
    - links and calculating, 43-24
    - mathematical operations, 43-25
    - overview, 43-6
    - overwriting local values, 43-26
    - recovering from errors, 43-18
    - reporting, 43-29
    - running, 43-26, 43-27, 43-30
    - saving outline changes, 43-17, 43-23
    - suppressing in reports, 36-14
      - tagging dimensions for, 9-10
  - currency database (defined), 43-3
  - currency database types, 43-6
  - currency fields, 20-4
  - Currency Name dimension (example), 43-5
  - currency name field type
    - in header records, 14-24
    - in rules files, 14-15
    - nulls and, 13-7, 13-10
  - currency names, 43-3, 43-5, 43-12
    - defining, 14-15, 43-12
  - Currency page (Database Information), 46-12
  - Currency page (Database Settings), 41-7, 43-25
  - Currency Partition dimension, 43-4, 43-26, 43-27
    - creating, 43-15
    - usage overview, 9-12
  - currency partition dimension
    - description, 5-20
  - currency partitions, 43-4
    - caution for transparent, 43-26
    - creating, 43-15
  - currency sample databases, 43-3
    - linking, 43-24
  - currency symbols, 20-4
  - Currency Time dimension (example), 43-5
  - Currency Type dimension (example), 43-5
  - @CURRMBR function, 26-36
  - @CURRMBRRANGE function, 26-36
  - customizing
    - applications, 1-5, 45-13
    - Data Prep Editor, 21-4
    - file delimiters, 21-11
    - Hyperion Essbase kernel, 41-2, 41-3
    - Outline Editor, 8-24, 8-25, 8-26, 8-27
    - page layouts, 36-6, 36-18
      - examples, 37-23, 37-27
    - user interface, 1-5
    - views, 8-25
  - cutting text
    - Calc Script Editor, 31-41
    - Formula Editor, 26-18
    - Report Editor, 35-20
- ## D
- dashes (–)
    - in dimension and member names, 8-15
    - in member names, 20-5
    - in names in scripts and formulas, 8-16
    - in report scripts, 36-6

- data, 1-3, 20-15
  - accessing. *See* access
  - archiving, 48-2
  - calculated vs. input, 5-25
  - categorizing. *See* dimensions
  - centering in reports, 36-9, 36-17
    - example, 37-27
  - changing, 6-11, 22-22, 26-48
    - prerequisite for, 17-37
  - clearing, 6-18, 12-3, 22-23, 31-53, 34-19
    - in calculated columns, 36-19, 36-20
  - computing relationships, 3-4
  - controlling flow, 6-8
  - copying, 6-18, 29-20, 31-54, 34-3, 34-19
  - corrupted, 40-8, 48-2
    - operations causing, 48-5
  - derived, 6-13
  - developing logical structures for, 3-13
  - distribution characteristics, 4-2
  - entering in data sources, 20-4, 20-5
  - exporting, 29-21, 39-1, 39-13
    - for backups, 48-3
    - into other forms, 36-15
    - reporting example, 37-37
  - importing, 39-1, 39-16
  - improving access to, 6-2, 6-8
  - in targets, 16-41
  - irrelevant, 5-15
  - loading, 23-13
    - aborted transactions and, 42-12
    - description, 23-8, 23-12
    - dynamic calculations and, 29-21
    - for testing purposes, 5-25
    - from external sources, 5-17, 20-3, 20-8, 23-1
    - from partitioned applications, 6-18, 6-20, 16-5
    - from rules files, 20-10
    - getting information about, 46-17
    - incrementally, 33-3
    - missing values and, 33-38
    - optimizing, 24-7, 24-9, 24-10, 24-12
    - overview, 7-3, 20-1, 20-2, 20-8, 24-7
    - prerequisites, 17-38, 23-2
    - restrictions, 20-5, 20-6, 20-15, 20-17
    - supported formats, 23-2
- data (*Continued*)
  - tips, 23-17, 23-18
  - troubleshooting problems with, 24-1
  - unauthorized users and, 13-46
  - managing, 40-4
  - manipulating remote, 6-15
  - missing, 16-62
  - not associated with members, 5-21
  - organizing, 3-13
  - partitioning guidelines, 6-3, 6-8, 6-9
  - pivoting, 3-2
  - protecting, 42-1, 48-1
  - recalculating, 3-19, 29-20
    - after exporting, 48-4
    - after member name changes, 8-15
    - after repositioning members, 8-20
    - Dynamic Calc And Store members, 29-3
    - examples, 32-5
    - for Personal Essbase servers, 39-12
    - in sparse dimensions, 31-50, 34-11
    - Intelligent Calculation and, 34-4, 34-6
    - two-pass calculations and, 33-26
  - referencing in dimensions, 3-8, 33-6, 33-7
  - refreshing, 6-13
  - reloading exported, 48-4
  - replicating, 6-8, 6-13
  - reporting combinations, 37-14, 37-17
  - retrieving, xliii
  - sharing, 6-4, 6-6
    - across multiple sites, 6-9, 13-44
    - disabling, 5-22, 9-20, 9-24, 14-25
    - in partitioned databases, 16-8, 16-22, 16-33
    - sorting for reports, 35-7, 36-45, 36-46, 36-49
    - examples, 37-27, 37-32, 37-51
  - storing. *See* storage
  - synchronizing, 6-2, 6-8, 6-36
    - partitioning and, 40-21
  - time-sensitive, 5-16
  - transactions and redundant, 41-16, 42-11
  - validating, 24-3
  - viewing, 3-2, 3-8, 3-11
    - in data targets, 6-23
    - with Data Prep Editor, 21-3, 21-4

- data analysis, 3-2
  - defining objectives, 5-6
  - example, 5-11
  - getting started tips, 2-2
  - optimizing, 12-1
  - single-server applications, 5-4, 5-6
- data blocks, 4-4, 4-6, 28-1
  - accessing locked, 20-24, 33-23
  - as multidimensional arrays, 4-6
  - calculating dirty, 34-14
  - calculating values in, 28-24, 34-12
    - caution for partial calculations, 31-54
    - defining calculation order, 28-11
  - calculations and, 33-12
  - categorizing, 28-3
  - checking characteristics of, 46-13
  - checking structural integrity, 42-13
  - clearing values, 31-53, 34-19
  - compressing, 40-8, 40-10, 40-11
  - creating from Intelligent Calculations, 34-10
  - defined, 4-4
  - exporting, 48-4
  - getting size, 4-8
  - locking, 33-22, 33-23, 42-5
    - with committed access, 42-7, 42-9
    - with uncommitted access, 42-6
  - managing, 40-5
  - marking as clean/dirty, 34-2, 34-6
  - ordering, 28-3
  - removing, 29-20
  - renumbering, 28-14
  - restructuring, 40-15, 40-18, 40-28
  - retrieving, 4-5, 4-11, 4-12
  - setting transactions for, 41-16
  - size considerations, 15-5, 15-6, 33-2, 41-29
  - storing, 4-8
  - two-dimensional example, 4-14
  - updating, 34-1
  - viewing locks, 17-37
  - with no values, 4-7
- data cache, 40-5
  - as storage unit, 15-2
  - defined, 40-5
  - determining size, 15-15
- data cache (*Continued*)
  - setting size, 15-16, 33-14
    - with Application Manager, 41-11
    - with ESSCMD, 41-12
    - with Hyperion Essbase kernel, 41-9
  - updating, 41-10, 41-12
- data cells. *See* cells
- data compression, 4-7
  - checking compression ratio, 40-11
  - enabling/disabling, 40-10
  - estimating block size, 15-5
  - fragmentation allowance, 15-10, 40-14
  - optimal configuration, 40-10
  - overview, 40-11
  - processes described, 40-8
  - ratio, 46-14
  - repetitive values, 40-8, 40-9, 40-10
  - specifying, 41-27, 41-28
  - types defined, 40-8
  - updating, 41-29
  - values (ESSCMD), 41-28
  - viewing settings information, 46-11
- data consistency. *See* data integrity
- data coordinates, 3-8
- data extraction, 38-7, 39-1
  - characters ignored, 36-3
- data extraction commands, 35-9, 36-27
  - defined, 36-1
- data fields
  - defined in data source, 20-2
  - defining columns as, 22-21
  - entering data, 20-4
  - formatting, 20-15, 20-16
  - in data sources, 20-2, 20-3
  - invalid, 20-6
  - reversing values in, 22-27
  - with no values, 20-16, 22-15
- data file cache
  - as storage unit, 15-2
  - defined, 40-5
  - setting size
    - with Application Manager, 41-10
    - with Hyperion Essbase kernel, 41-9
  - updating, 41-10

- Data File Properties dialog box
  - defining header information, 21-14
  - mapping fields, 22-4
  - setting file delimiters, 21-10
  - undoing field operations, 22-11
- data files
  - See also* files
  - caution for recovery and, 42-11
  - cross-platform compatibility, 47-16
  - estimating size, 15-3, 15-5
  - opening, 21-5, 21-7, 21-8
  - replicating, 48-1
  - restructuring, 40-15, 40-27
  - spanning multiple volumes, 40-5
  - storage allocation and, 40-12
  - storing, 41-20
- data filters, 18-2
  - assigning, 18-15
  - copying, 18-12
  - creating, 6-30, 6-31, 18-3, 18-6
  - defining in Application Manager, 18-3, 18-5
  - editing, 18-11
  - inheriting, 18-2, 18-15, 18-16
  - overlap conflicts, 18-16, 18-17
  - overriding, 16-5, 25-12
  - overview, 18-1, 18-2
  - removing, 18-14
  - renaming, 18-13
  - restrictions on applying, 18-15
  - saving, 18-2
  - viewing existing, 18-3
  - write access to database, 13-46
- data hierarchies, 3-4
  - relationships defined, 3-5, 3-6
- data integrity, 40-8, 42-2
  - checks failing, 42-13
  - committed access and, 42-2
  - maintaining, 17-37
  - retaining duplicate data, 42-11
  - uncommitted access and, 42-3
  - validating, 42-13
- data load
  - fields, 21-4
  - free-form, 20-18
  - load rules in the source data, 14-23
- data load (*Continued*)
  - mode, 21-3
  - parent members, 23-17
  - parents and missing values, 33-38
  - rules
    - creating, 20-9
    - defined, 20-8
    - described, 7-3
    - when to use, 20-8
- Data Load Completed dialog box, 23-13, 23-14, 23-16
- Data Load dialog box, 23-8
  - building dimensions, 14-26
  - changing outlines, 23-11
  - selecting data sources, 23-4, 23-5
- Data Load Properties page
  - naming fields, 22-13
- Data Load Rules button (Application Manager), 7-8
- Data Load Settings dialog box
  - changing data values, 22-22
  - clearing existing values, 22-24
  - defining header information, 21-12
  - defining selection criteria, 21-19
  - flipping field values, 22-27
- Data Manager, 40-2
  - overview, 40-5
- data points. *See* cells
- Data Prep Editor
  - building dimensions, 14-2, 14-10, 14-20, 14-21
  - creating rules files, 13-2, 14-2
  - customizing, 21-4
  - defining header information, 21-11
  - dimension building mode, 14-12
  - displaying
    - dimension building fields, 14-3
    - records, 21-20
  - enabling/disabling toolbar, 21-4
  - loading data sources, 21-6, 21-7, 21-8
  - opening, 14-3, 21-3
  - selecting data load or dimension build view, 21-4
  - validating rules files, 14-12, 14-21
  - viewing rules files, 21-2
  - window, 14-12
- data redundancy overhead, 15-11, 15-12
- Data Replication dialog box, 16-61
- data repositories, 5-3

- data sets, 4-8
  - copying portions, 6-10
  - distribution, 4-2
  - non-typical, 4-13
- data sources
  - accessing data, 6-10
    - replicated partitions and, 6-10
  - adding headers, 14-23, 14-24, 21-11, 21-13
  - adding records, 13-5, 13-8, 13-11
  - altering to build dimensions, 14-23
  - appending information to, 14-23
  - associating with rules files, 23-9
  - building dimensions, 13-3, 13-5, 13-8, 13-11, 13-12
  - changing outlines, 16-48
  - copying, 47-12
  - creating outlines with, 13-2
  - creating shared roll-ups from multiple, 13-44
  - debugging, 14-30
  - defined, 6-3
  - defining, 13-4
    - for multiple partitions, 6-4
    - for replicated partitions, 16-6
    - for transparent partitions, 16-21, 16-32
  - entering data, 20-4, 20-5
  - field types, 20-2
    - described, 20-3, 20-5
    - invalid, 20-6
  - fixing
    - no loads, 23-16
    - partial loads, 23-14
  - for dimension builds, 13-1
  - formatting, 20-12, 20-14, 20-15, 20-17, 20-18
    - overview, 20-2
  - free-form, 20-18, 20-19, 20-22
    - optimizing, 24-11
  - identifying, 5-4
  - load failing, 24-4
  - loading, 13-2, 14-27, 20-8, 24-11
    - in Data Prep Editor, 21-6, 21-7, 21-8
    - prerequisites, 23-2
    - troubleshooting problems, 24-1, 24-6
  - logging into, 16-7
  - losing connections to, 16-62
- data sources (*Continued*)
  - mapping information, 6-6
  - mapping members, 16-42
    - linked partitions, 16-35, 16-37
    - replicated partitions, 16-11, 16-17
    - transparent partitions, 16-25, 16-27
  - member names differing from targets, 6-6
  - members with no ancestor specified, 13-12
  - missing members, 16-43, 20-13
  - numbering members, 13-5
  - opening, 21-5
  - opening in Data Prep Editor, 14-12
  - optimizing, 24-11, 24-12
  - ordering
    - columns, 13-11
    - fields, 22-5, 22-11
    - records, 13-3, 13-5, 13-8
  - overview, 20-1, 20-2, 21-11
  - partitioning information for, 6-6
  - propagating outline changes, 16-6, 16-21, 16-32
    - process summarized, 16-47, 16-48
  - remote data retrievals and, 6-15
  - removing members, 14-11
  - replicating derived data, 6-13
  - replicating members, 6-13
  - replicating partial sets, 6-10
  - selecting, 6-9, 16-4
    - how to, 21-1
    - SQL, 23-4
    - using Application Manager, 23-3
    - using Windows, 23-7
  - selecting text or spreadsheet, 23-5
  - setting member properties, 14-24
  - specifying
    - field type, 13-5
  - specifying shared areas, 16-8, 16-22, 16-33
  - supported, 23-2
  - unavailable, 24-1, 24-4
  - updating changes to, 6-15, 16-60
  - validating rules files from, 14-21
  - viewing, 21-3, 21-4
    - with different values, 22-25
    - with new member lists, 13-12
- Data Storage dialog box, 4-8, 8-9

- data storage property
  - setting, 14-6
- data storage. *See* storage; Hyperion Essbase kernel
- data subsets
  - calculating, 31-12, 31-54, 31-55
    - example, 32-3
    - with Intelligent Calculation, 34-4
  - clearing, 31-53
  - copying, 31-54
  - copying to Personal Essbase, 39-2
    - prerequisites, 39-7
  - loading, 34-1, 39-9
- data targets
  - accessing data, 6-10, 6-18
  - archiving, 6-18
  - calculations and, 6-14, 6-21
  - changing data in, 6-11
  - changing outlines, 16-48
  - copying from, 6-10
  - defined, 6-3
  - defining
    - for multiple partitions, 6-4
    - for replicated partitions, 16-6
    - for transparent partitions, 16-21, 16-32
  - logging into, 16-7
  - losing connections to, 16-62
  - mapping information, 6-6
  - mapping members to data sources
    - linked partitions, 16-35, 16-37
    - replicated partitions, 16-11, 16-17
    - specifying specific areas, 16-42
    - transparent partitions, 16-25, 16-27
  - member names differing from source, 6-6
  - missing data, 16-62
  - partitioning information for, 6-6
  - propagating outline changes, 16-6, 16-21, 16-32
    - process summarized, 16-47, 16-48
  - selecting, 16-4
  - specifying shared areas, 16-8, 16-22, 16-33
  - updating changes to, 6-11, 16-5, 16-60
  - viewing data in linked partitions, 6-23
- data values, 3-8, 6-2, 25-1, 35-8
  - See also* missing values; range of values
  - accumulating, 26-45
  - data values (*Continued*)
    - assigning
      - to member combinations, 26-46
      - to variables, 26-48
    - averaging
      - for time periods, 9-6, 30-1, 30-4
      - non-zero values and, 30-5
      - reporting examples, 37-32, 37-47
      - with formulas, 26-38, 27-3
    - changing, 22-22
    - comparing, 3-11, 9-9, 38-5
    - compression and repetitive, 40-8, 40-9, 40-10
    - defined, 3-9
    - displaying specific, 4-6
    - distribution among dimensions, 4-2
    - duplicating, 5-22
    - dynamically calculating, 29-2, 29-6, 29-15
    - entering in empty fields, 22-15
    - filtering, 18-2
    - flipping, 22-27
    - formatting in reports, 36-23, 36-26
      - examples, 37-17
    - identical, 5-28
    - inconsistent, 40-8
    - incorrect, 24-2
    - interdependent, 26-33
    - iterating through, 26-44
    - looking up, 26-4
    - measuring, 5-20
    - member with no, 9-22
    - negative, 20-4
      - flipping, 22-27
      - variance as, 26-39
    - nulls, 13-7, 13-10
    - optimizing in sparse dimensions, 24-8
    - ordering in reports, 36-45, 36-46
      - example, 37-55
    - out of range, 20-20
    - overwriting, 22-23, 23-17, 30-6
      - for currency conversions, 43-26
    - placing retrieval restrictions, 36-46, 36-50
      - example, 37-54
    - reading multiple ranges, 20-22
    - referencing, 3-8, 25-6, 26-49, 33-6, 33-7

- data values (*Continued*)
  - retrieving
    - dynamically calculated, 29-4, 29-10, 29-16, 29-21
    - from other databases, 26-45
    - from remote databases, 6-15, 29-14, 29-25
  - retrieving for reports, 35-6
    - setting maximum rows allowed, 36-50
    - with conditions, 36-45, 36-48, 37-54, 37-55
  - rolling up, 5-26
  - rounding, 24-12, 26-38
  - scaling, 22-25
  - setting maximum number of, 3-14
  - storing, 9-20, 9-23
  - temporary, 31-12, 32-18
  - truncating, 26-39
  - unable to change, 16-5
  - unexpected, 40-8
  - unique, 4-5
    - assigning #MISSING values to, 33-35
    - Intelligent Calculation and, 34-10
  - unknown, 20-15
  - variables as, 7-14
  - void, 16-43
- Data Values page (Data Load Settings)
  - adding to or subtracting from existing values, 22-22
  - flipping field values, 22-27
- Database Access dialog box, 17-28
- Database Access options (Database Settings), 17-35
- database administrators. *See* administrators
- database cells, 3-8
  - accessing, 4-6
    - simultaneously in different databases, 6-23
  - annotating, 12-2
  - copying range of, 31-54
  - determining calculation order for, 28-14, 28-22
    - examples, 28-15, 28-16, 28-18, 28-20
  - empty, 4-7
    - caution for storing, 4-16
  - linking objects to, 12-1
  - mapping to targets, 6-3
  - ordering in blocks, 4-6
- database cells (*Continued*)
  - partitioning and, 6-12, 6-17, 6-23
  - removing linked objects, 12-6
  - returning unique values for, 4-5
  - shaded, 3-9
- Database Copy dialog box, 47-9
- database design
  - attribute dimensions, 5-15
- Database Designer privilege. *See* DB Designer privilege
- database directory, 47-5
- database files, 47-16
- Database Filter layer, 18-1, 18-2
- Database Information dialog box, 15-4, 33-2, 46-3, 46-9
  - performance checks, 15-18
- Database Locks dialog box, 17-37
- database outlines. *See* outlines
- Database Settings dialog box
  - aggregation of missing values, 33-37
  - calc options settings, 25-7
  - changing Hyperion Essbase kernel settings, 41-3, 41-5
  - global database access, 17-34
  - scope of storage settings, 41-4
  - setting
    - currency conversion links, 43-25
    - retrieval buffer size, 29-12, 38-3, 38-4
  - settings described, A-2
  - two-pass calculations, 33-27
- databases
  - See also* data; partitioned databases
  - accessing, 1-4, 6-30, 17-24, 17-34
    - remote, 6-8
  - analytical vs. transaction processing, 3-1
  - annotating, 7-13
  - associating
    - with calc scripts, 31-19, 31-25, 31-28, 31-43
    - with reports, 35-26
  - associating index files with, 46-16
  - attaching to. *See* connections
  - backing up, 48-1

databases (*Continued*)

- building, 3-3, 3-4, 7-9
  - development process for, 5-4
  - example for, 5-3
  - prerequisites for, 5-4
- calculating multiple, 31-58
- checking
  - operational status, 39-4
  - which databases are currently running, 46-2
- checklist for analyzing, 5-17
- closing related application simultaneously, 45-9
- configuring, 33-2
- contents for typical, 3-16
- copying, 7-9, 47-9
- creating, 4-16, 5-16, 7-9
  - for Personal Essbase, 39-3
  - process summarized, 7-5
  - with Application Manager, 7-11
- creating accounts for, 6-26, 6-31, 6-32
- currency conversions and, 43-2
- deleting, 47-11
- determining scope, 5-11
- displaying statistical information, 33-2
- distributing. *See* partitioning
- fine tuning, 5-10
- identifying data sources, 5-4
- implementing global security, 17-29, 17-30, 17-32, 17-34
- linking related, 6-33, 6-38
- loading applications simultaneously, 45-8
- minimizing downtime, 6-13
- mission-critical, 6-8
- multidimensional defined, 3-1
- naming rules, 7-12
- navigating between, 6-23, 6-25
- non-contiguous portions in, 6-6
- opening, 7-7, 7-8
- optimizing access, 6-2, 6-8
  - to large, 13-1
- optimizing retrieval, 29-10
- partitioning, 6-3, 6-7
  - guidelines for, 6-8, 6-9
  - sample applications showing, 6-10
- planning prerequisites, 5-4

databases (*Continued*)

- processing requests, 1-4, 45-13
- protecting data, 42-1, 48-1
- rebuilding, 42-13
- reducing size, 6-13
- related information among, 6-8
- reloading, 47-20
- removing
  - partitions, 16-58
- removing dimensions, 3-19
  - restructuring and, 40-29
- renaming, 47-10
- restructuring, 40-14, 40-21
  - changing outlines and, 8-15, 8-20
  - dynamic calculations and, 29-23
  - immediately, 40-19, 40-22
  - in partitioned applications, 6-19
  - Intelligent Calculation and, 34-3, 34-19
  - process described, 40-16, 40-20
- sample, xlv, 43-1
- selecting, 7-7
  - with ESSCMD, 44-7
- size determinations, 3-14, 15-1, 15-8, 15-9
- slicing, 3-11
- splitting, 5-16, 6-2
- starting, 7-6
  - from Server Agent, 45-8
- stopping, 7-6, 45-8, 45-9
- testing design, 5-25
- updating, 13-46, 17-37
- viewing
  - in Application Manager, 21-2
  - information about, 46-2, 46-9
  - with differing dimensionality, 6-2
- Databases list box, 7-7
- DATACOPY command, 29-20, 33-11
  - currency conversions and, 43-27
  - partitioned applications and, 6-18
  - usage overview, 31-54, 34-19
- DATAERRORLIMIT setting, 23-18, 24-2
- DATALOAD.ERR
  - See also* error log files
  - renaming, 24-5
  - setting, 23-12
  - using to debug data load problems, 24-1

- DATALOAD.TXT, 24-5
- date
  - attribute dimensions
    - changing the member name format, 10-22
    - tagging, 9-14
  - attributes
    - defined, 10-6, 10-7
    - duplicate values, 10-16
- date calculations, 5-30, 30-7
  - examples, 28-5
- dates, in calculation formulas, 26-46
- day-to-date calculations, 30-9
- .DB files, 47-2
- DB Designer privilege, 17-26, 17-30, 17-36
- dBASE data files, 47-2
- dBASE databases. *See* SQL databases
- dBASE index files, 47-2
- .DBB files, 47-2
- .DBF files, 47-2
- .DDB files, 16-41, 16-48, 47-2
- .DDM files, 47-2
- .DDN files, 47-2
- deadlocks, 42-9
- deallocation (storage), 41-22, 41-26
- debugging data loads, 24-1
- debugging tool, 23-12
- DECIMAL command, 36-10, 36-12, 37-18
- decimal points, 20-4
- declaring
  - arrays, 32-8
  - variables, 31-12, 31-52, 32-18
- @DECLINE function, 26-45
- Default table (aliases), 11-3
  - updating during builds, 14-21
- defaults
  - aggregating missing values, 33-36
  - application settings, 17-33
  - cache size, 15-16
  - calculations, 33-27
    - setting, 25-7, 34-3, 34-5
  - calculator cache, 33-15
  - change log file size, 46-39
  - compression, 41-27
    - changing, 40-10
  - currency type, 43-25
- defaults (*Continued*)
  - data storage, 5-21, 8-9, 9-20
  - dimension properties, 9-2
  - dynamic builds, 14-11
  - error log file locations, 46-27
  - for database settings, A-2
  - Intelligent Calculation, 34-2
  - login information, 16-7
  - maximum lock time, 17-37
  - member selection for reports, 36-44
  - members names, 8-24
  - retrieval buffers, 38-2, 38-4
  - time balance tags, 9-5
  - variance reporting tags, 9-9
- Define Attribute Dimensions dialog box, 14-5
- Define Filter dialog box, 18-5
- Define SQL dialog box, 21-9
- defining
  - See also* adding; creating; setting
  - a single data value, 26-47
  - area-specific mappings, 16-46
  - calc script as default, 25-7
  - calculation order, 28-1
    - cells, 28-14, 28-22
    - data blocks, 28-11
    - dimensions, 28-6
    - dynamically calculated values, 29-16, 29-17
    - forward references and, 28-8
    - members, 28-4, 28-5
  - calculations (checklist for), 5-36
  - columns as fields, 22-21
  - consolidation paths, 5-26, 5-28
    - checklist for, 5-28
  - currency conversions, 43-2
  - currency names, 43-12
  - custom attributes, 9-25
  - data sources, 13-4
    - for multiple partitions, 6-4
    - for replicated partitions, 16-6
    - for transparent partitions, 16-21, 16-32
  - data storage properties, 9-20
  - data targets
    - for multiple partitions, 6-4
    - for replicated partitions, 16-6
    - for transparent partitions, 16-21, 16-32

defining (*Continued*)

- dimension properties, 5-19, 9-1, 9-2, 14-6
  - caution for two-pass tags, 9-15
- dimensions as flat, 33-4, 33-22
- dynamic calc properties, 9-21
- field type, 13-5, 14-12
- filters in Application Manager, 18-3, 18-5
- global access levels, 17-31, 17-34, 17-35
- member properties, 9-1, 9-16, 9-18, 14-24
  - as label only, 9-22
- members as label only, 14-25
- multiple layout reports, 37-36
- page layouts, 36-3, 36-4, 36-7
- parent/child relationships, 13-11
- partitioned areas
  - linked partitions, 16-33, 16-34
  - replicated partitions, 16-8, 16-9, 16-10
  - transparent partitions, 16-22, 16-23
- partitions, 6-3, 6-8, 6-9
  - linked, 6-25, 6-26
  - multiple, 6-4
  - replicated, 6-11, 6-12, 6-13
  - transparent, 6-16, 6-18
  - with Partition Wizard, 16-3, 16-4, 16-20, 16-31
- selection/rejection criteria, 21-15, 21-16
  - multiple for single field, 21-16, 21-17
  - on multiple fields, 21-18
- shared member properties, 9-23, 9-25
- sort order, 36-46
- two-pass calculations, 9-15, 14-25
- UDAs, 9-25

definition files. *See* partition definition files

delays, 33-24

Delete Alias Table dialog box, 11-10

DELETEAPP command, 47-8

Deleted Dimensions dialog box, 40-30

DELETEDDB command, 47-11

DELETEDGROUP command, 17-20

DELETELOCATION command, 7-22

DELETELOG command, 46-30, 46-33

DELETEUSER command, 17-20

DELETEVARIABLE command, 7-18

## deleting

- See also* clearing
- aliases, 11-6
- applications, 17-35, 47-8
- calc scripts, 31-35, 31-36
- contents from alias tables, 11-11
- data blocks, 29-20
- databases, 47-11
- dimensions, 3-19
  - restructuring and, 40-29
- filters, 18-14
- formulas, 26-14
- items from outlines, 3-19, 11-10
- linked objects, 12-6
- locks, 17-35, 17-37, 46-40, 47-15
  - automatically, 17-37
- log files, 46-29, 46-32
- members from data sources, 14-11
- objects, 47-13
- partitions, 16-58
- substitution variables, 7-17
- temporary files, 40-17
- text
  - in calc scripts, 31-41
  - in formulas, 26-18
  - in report scripts, 35-20
- user groups, 17-20
- users, 17-11, 17-19
- white space in fields, 22-18

delimiters

- See also* file delimiters
- commas in numbers, 20-4
- exports/imports, 39-13, 39-16
- member lists, 26-35
- report script commands, 36-2, 36-15

Demo application, xliv

Demo Basic database, xliv
 

- sample reports, 37-1

dense dimensions, 4-2

- See also* dimensions; sparse dimensions
- attribute design approach, 10-13
- calculating values in, 28-22, 28-23, 31-56
- examples, 28-15, 28-17, 28-18

- dense dimensions (*Continued*)
  - defined, 4-2
  - dynamically calculating, 29-7, 29-15, 29-16, 29-23
  - implications for restructuring and, 40-15, 40-18
  - Intelligent Calculation and, 34-10, 34-12
  - location in outline, 5-23, 10-14
  - member order and, 4-6
  - partitioning, 6-14, 6-20
  - referencing values in, 33-11
  - reporting on, 38-7
  - selecting, 4-8
  - selection scenarios, 4-11, 4-12, 4-13, 4-14
  - setting, 14-7
  - storing, 8-9
    - viewing member combinations, 4-4, 4-7
- density, 33-2
- depreciation, 1-2, 26-45, 26-46
- derived data, 6-13
- DESC command
  - described, 36-47
  - usage example, 37-55
- @DESCENDANTS function, 16-9, 26-36
- descendants
  - checking for, 26-30
  - clearing, 22-25
  - currency conversions and, 43-9
  - defined, 3-6
  - moving, 14-10
- DESCENDANTS command
  - entering in report scripts, 36-44
  - usage, 36-28
- descending sort order, 36-44
  - applying to output, 36-47
  - members in outlines, 8-21
- design checklists
  - analyzing database scope, 5-17
  - creating business models, 5-10
  - defining calculations, 5-36
  - defining consolidations, 5-28
  - defining dimension properties, 5-22
  - identifying data sources, 5-5
  - partitioning databases, 6-8, 6-9
  - selecting partition types, 6-27
- design guidelines
  - attributes, 10-12
  - dimensions, 5-10
  - outlines, 5-23
- designing
  - See also* building; creating
  - for optimal calculations, 10-14, 33-2
  - partitioned applications, 6-1, 6-2, 6-8, 6-9, 6-10
    - scenarios for, 6-33, 6-36, 6-38
  - reports, 35-7, 35-10
  - simple applications, 3-12, 3-13, 3-16
  - single-server applications, 5-1, 5-4
- Desktop window. *See* Application Desktop window
- detail members, 3-6
- developing applications, 1-2, 3-8
  - operations not supported, 3-2
  - process summarized, 5-2
- diagnostic tools, 46-1
  - overview, 46-2
  - reference table, 46-18
- dialog boxes
  - informational, 46-2
  - quick reference, 46-18
  - refreshing pages in, 46-3
- differences
  - between attributes and UDAs, 10-9
  - between standard and attribute dimensions, 10-7
- DIMBOTTOM command
  - entering in report scripts, 39-14
  - usage, 36-28
- DIMBUILD.ERR, 14-30, 24-5
- dimension building rules
  - summary for attribute dimensions, 13-30
- dimension build
  - See also* dynamic builds
  - associating
    - aliases with attributes, 13-26
    - base dimension members with attributes, 13-19
  - attribute members options, 14-11
  - defining
    - attribute dimensions, 14-4
    - standard dimensions, 14-4
  - fields, 21-4
  - mode, 13-2
  - position of fields in rules file, 13-22

- Dimension Build Completed dialog box, 14-29, 23-13
- Dimension Build Settings dialog box, 14-9
  - adding members, 13-14, 13-16, 13-17
  - creating rules files, 13-3, 14-2
  - naming dimensions, 14-3
  - setting properties, 14-20
- Dimension Build Settings page (Dimension Build Settings), 14-9
- dimension building
  - field types, 14-15
  - rules, 7-3
  - setting mode in Data Prep Editor, 14-12
- Dimension Building Properties page (Field Properties), 14-14
- Dimension Definition page (Dimension Build Settings), 14-3
- dimension fields
  - formatting, 20-12
  - in data source, defined, 20-2
- dimension names
  - maximum length, 8-15
- Dimension Properties dialog box, 14-6
- Dimension Properties page (Dimension Properties), 14-7
- Dimension Specification dialog box
  - adding comments, 9-27
  - applying time balance tags, 9-9
  - defining dimension type, 9-3, 9-4, 9-12, 9-14, 43-15
- dimension type
  - setting, 14-6
- dimensions, 1-3, 3-3
  - See also* dense dimensions; sparse dimensions; attribute dimensions; standard dimensions; base dimensions
  - adding comments to, 9-27
  - adding members, 8-19, 9-23, 13-5, 13-13, 13-15, 13-16
  - example for, 3-13, 3-16
  - for currency conversions, 43-22, 43-25
  - guidelines for, 5-27
  - partitioned applications, 6-13
  - restrictions, 3-5
- dimensions (*Continued*)
  - adding to outlines, 3-18, 4-8, 4-11, 8-15, 8-18
    - performance considerations, 33-3, 33-8
    - restructuring and, 40-28
  - applicable to business models, 5-6
  - arranging in hierarchies, 3-4, 3-5
  - associating formulas with, 9-28
  - attribute
    - See also* attribute dimensions
    - described, 10-4
  - auto-configuring, 4-8, 8-9, 14-21
  - base
    - See also* base dimensions
    - defined, 10-4
  - building, 23-8, 23-10, 23-12, 23-13
    - dynamically *See* dynamic builds
    - example for, 3-18
    - guidelines for, 5-7, 5-12
    - prerequisites, 21-15, 21-16, 23-2
    - restructuring databases and, 40-14
    - with dynamically calculated members, 29-23
    - with rules files, 13-2
  - calculating members in, 28-4, 28-5
  - calculating series of, 31-51
  - calculator cache and, 33-16
  - categorized, 4-2
  - changing, 14-10, 40-27
    - default configurations, 8-9
    - for currency conversions, 43-8
  - changing properties, 13-4, 14-6, 14-10, 14-23
  - clearing member combinations, 22-25
  - consolidation levels, 3-7
  - creating filters for, 18-6
  - databases with different, 6-2
  - defined, 3-3
  - defining as flat, 33-4, 33-22
  - defining properties, 5-19, 9-1, 9-2, 14-6
    - caution for two-pass tags, 9-15
  - deleting, 3-19
    - restructuring and, 40-29
  - determining optimal configuration, 4-9, 4-13
  - determining valid combinations, 5-13
  - displaying
    - in Calc Script Editor, 31-47
    - in Formula Editor, 26-24
    - in outlines, 8-3

- dimensions (*Continued*)
  - displaying information about, 46-13
  - examples
    - time-balanced data, 9-4, 9-5
  - fixing as constant, 3-12
  - getting member combinations, 4-7
  - getting started with setting up, 2-3
  - grouping in calc scripts, 31-50, 31-51, 33-13
  - handling missing values in, 9-6
  - identified in data source, 20-2
  - inheritance characteristics, 9-2
  - irrelevance across, 5-15
  - mapping fields to, 22-12
  - missing, 20-13
  - missing members, 37-45
  - moving in outlines, 8-22
  - naming, 8-15, 8-18, 14-3
  - naming levels in current, 14-8
  - nesting, 38-8
  - non-specific, 5-20
  - optimum order in outline, 10-14
  - ordering, 28-6
  - ordering members, 4-6
  - predefined types described, 5-20
  - referencing data in, 3-8, 33-6, 33-7
  - relationship among members, 3-6, 25-2, 28-4
  - renaming, 8-10
  - selection guidelines, 5-7
  - sharing members, 9-23, 13-32
  - single-server models, 5-22
  - sorting, 8-21, 14-11
  - splitting, 5-14
  - standard
    - See also* standard dimensions
    - alternative for attributes, 10-13
    - compared with attribute types, 10-7
    - described, 10-4
  - storage defaults, 8-9
  - tagging
    - as Currency Partition, 43-15
    - as specific type, 9-2
    - for currency conversion, 9-10
  - two-pass calculations and, 9-15
  - updating, 14-26
- DIMTOP command
  - usage, 36-28
- direct I/O, 40-2
- directories
  - application, 47-5
  - database, 47-5
  - default
    - for calc scripts, 31-26
    - for error logs, 24-1
    - for installation, 8-12
  - default for calc scripts, 31-28
  - dump, 46-27
  - error log files, 46-27
  - spreadsheets as data sources, 21-7
  - text files as data sources, 21-6
- dirty status, 34-11
  - calculating blocks with, 34-14
  - clearing data and, 34-19
  - copying data and, 34-19
  - currency conversion and, 34-20
  - Intelligent Calculation and, 34-4
  - marking blocks with, 34-2, 34-6
- Disabled Usernames dialog box, 17-41
- disabling member sort, 36-45
- discarding records, 21-16
- disconnecting
  - from server, 17-33
  - other users, 17-35, 17-38
- discontinuous field selection, 22-2
- @DISCOUNT function, 26-45
- discount, calculating, 26-45
- disk drives
  - copying applications to, 47-6
  - copying databases to, 47-9
  - viewing information about, 46-3, 46-7
- Disk Drives page (Server Information), 46-7
- disk I/O, reducing, 24-10, 24-11
- disk space
  - allocating, 40-4, 40-11
  - example, 41-24
  - availability for restructuring, 40-28

- disk space (*Continued*)
  - calculations for determining
    - compressed block size, 15-5
    - data file storage, 15-7
    - data redundancy overhead, 15-11
    - database overhead, 15-8
    - fragmentation allowance, 15-10
    - potential block size and number, 15-5, 15-6
  - checking, 5-2, 46-6
  - conserving, 6-8
  - estimating, 15-3, 15-12
    - for linked reporting objects, 15-13
    - for partitioned applications, 15-12
  - freeing, 45-8, 46-29, 46-32
  - general requirements, 15-3
  - optimizing restructures and, 40-18
  - out of, 48-6
  - partitioned databases, 6-8, 6-9
    - with replicated partitions, 6-13
    - with transparent partitions, 6-18
  - unused and fragmented, 40-14
- Disk Swapping option, 46-6
- disk volumes
  - allocation and, 40-4
  - caution for specifying name only, 41-21
  - data files and, 40-5
  - data storage and multiple, 40-11
  - deallocating, 41-22, 41-26
  - index files and, 40-3
  - specifying
    - with Application Manager, 41-22
    - with ESSCMD, 41-24
    - with Hyperion Essbase kernel, 41-20
  - updating storage settings, 41-26
- Disk Volumes setting, 40-3, 40-4, 40-11, 41-20
  - enabling, 40-13
- display options
  - data in reports, 36-22, 36-26
  - formulas, 8-25
  - headings in reports, 36-6, 36-14
  - outlines, 8-25
  - reports, 35-21
- DISPLAYALIAS command, 11-3
- displaying, 16-41
  - access settings, 17-27, 17-28
  - active filters, 18-3
  - active user groups, 17-27
  - aliases, 8-25, 11-6
    - in Calc Script Editor, 31-48
    - in Formula Editor, 26-25
  - application information, 46-2, 46-8, 46-28, 46-31
  - applications, 21-2
  - available ports, 45-5, 45-10
  - changes to outlines, 46-35
  - current users, 17-8, 45-4, 45-10
  - data, 3-2, 3-8, 3-11
    - in targets, 6-23
    - with Data Prep Editor, 21-3, 21-4
  - data sources, 21-3, 21-4
  - data targets, 16-41
  - database
    - information, 46-2, 46-9
    - statistics, 33-2
  - databases, 21-2
  - dimensions
    - in Calc Script Editor, 31-47
    - in Formula Editor, 26-24
    - in outlines, 8-3
  - dynamically calculated members, 29-6, 29-11
  - field operations, 20-10
  - formulas, 8-25, 26-13
  - inactive users, 17-41
  - informational messages, 33-4, 33-23
  - linked objects, 12-6, 12-8
  - locked data, 42-7
  - locks, 17-37
  - log files, 29-10, 29-11, 31-9, 46-28, 46-31
  - member combinations, 4-4, 4-7
  - member names in reports, 36-43
  - members in outlines, 8-3, 29-6
  - names for generations and levels, 8-24
  - partitions, 16-57, 16-58
  - records, 21-20
  - replace operations, 20-11
  - selection/rejection criteria, 20-12
  - Server Agent commands, 45-5
  - server information, 46-3

- displaying (*Continued*)
  - software version, 45-5, 45-12
  - specific values, 4-6
  - storage information, 4-8
  - text in Outline Editor, 8-26
  - unique values, 4-5
- distributed client-server model, 1-4
- distributed databases. *See* partitioning
- distribution (multidimensional models), 4-2
- dividing
  - databases, 5-16
  - fields, 22-9
- dividing applications. *See* partitioned databases; partitions; splitting, databases
- dividing databases, 6-2
- division
  - consolidation codes in data source, 14-25
  - consolidation property, 9-18
  - modulus operations, 26-38
- .DLL files, 47-4
- Do Not Share option, 14-11
  - dynamic builds and, 13-36, 13-39, 13-42, 13-44
- DOCS directory, 31-12, 35-5, 45-13
- documentation, xlii
  - additional resources, xlv
  - suggested audience, xli
  - typographical conventions, xlv
- documents, linking external, 12-2
- dollar values, 43-2
  - converting to US\$, 43-26
- double quotation marks (")
  - enclosing member names, 20-14, 22-13
  - in application and database names, 7-12
  - in dimension and member names, 8-15
  - in ESSCMD commands, 44-2
  - in formulas, 14-11, 26-6, 31-15
  - in header information, 14-24
  - in report scripts, 39-8
  - terms in scripts and formulas, 8-16, 8-17
- double slashes (//) in report scripts, 36-3
- downtime, 6-13
- drilling across, 6-23, 6-25, 6-33
  - facilitating, 6-32
- D-T-D time series member, 30-9, 30-14
- dummy
  - parents, 13-16
  - strings, 22-15
- DUMP command, 45-5
  - usage overview, 45-12
- DUMP directory, 46-27, 46-34
- DUPGEN. *See* duplicate generation field type
- DUPGENALIAS. *See* duplicate generation alias field type
- DUPLEVELEL. *See* duplicate level field type
- DUPLEVELELALIAS. *See* duplicate level alias field type
- duplicate
  - data, 42-11
  - generations, 13-33
  - members, 13-44
- duplicate generation alias field type
  - arranging in rules files, 14-17
  - in header records, 14-24
  - in rules files, 14-15
  - nulls and, 13-7
- duplicate generation field type
  - arranging in rules files, 14-17
  - creating shared members, 13-33
  - in header records, 14-24
  - in rules files, 14-15
  - nulls and, 13-7
- duplicate level alias field type
  - arranging in rules files, 14-17
  - in header records, 14-24
  - in rules files, 14-15
  - nulls and, 13-10
- duplicate level field type
  - arranging in rules files, 14-17
  - in header records, 14-24
  - in rules files, 14-15
  - nulls and, 13-10
  - sharing members, 13-40, 13-41
- duplicate members
  - See also* shared members
- duplicating
  - See also* copying; replicating
  - data, 6-8
  - data values, 5-22
  - outlines, 8-4

## dynamic builds

- See also* dimension build
- adding new members
  - as child of specified parent, 13-16
  - to dimensions with similar members, 13-13
  - to end of dimensions, 13-15
- arranging fields, 14-17
- batch processing, 14-26
- bottom-up ordering, 13-3, 13-8, 13-10
- creating shared members, 13-40, 13-42
  - for different generations, 13-38, 13-39
  - for same generation, 13-33, 13-35, 13-36
  - with rule files, 13-32
- generation references and, 13-5
- global options, 14-20
- introduction, 13-1, 13-3
- level references and, 13-8
- members without specified ancestor, 13-12
- parent/child references and, 13-11, 13-44
- process summarized, 14-31
- restrictions, 13-46
- selecting build method, 13-4, 14-9
- selecting field types for, 14-12
- top-down ordering, 13-5, 13-7
- validating rules files, 14-21
- with Data Prep Editor, 14-2, 14-10, 14-20, 14-21
- with data sources, 13-3, 13-5, 13-8, 13-11, 13-12, 14-23
- with rules files, 13-2

## Dynamic Calc And Store members, 29-3

- adding to calculations, 29-22
- adding to formulas, 29-9
- clearing values, 31-53
- creating, 9-21
- currency conversions and, 29-20
- dense dimensions and, 29-7
- loading data into, 20-5
- partitioning, 6-14, 6-22
- replicating, 6-14
- reporting on, 38-8
- retrieving values, 29-4
- selecting, 29-13, 29-14, 29-15
- specifying in data source, 14-25
- viewing, 29-11

## Dynamic Calc And Store property

- applying, 29-7, 29-18, 29-22
- described, 5-21, 9-20
- removing, 29-23

## Dynamic Calc members, 29-2

- adding to calculations, 29-22
- adding to formulas, 29-8
- Attribute Calculations dimension, 10-31
- creating, 9-21
- currency conversions and, 29-20
- loading data into, 20-5
- partitioning, 6-13, 6-14, 6-22
- replicating, 6-13, 6-14
- reporting on, 38-8
- retrieving values, 29-4
- selecting, 29-13, 29-14, 29-15
- specifying in data source, 14-25
- viewing, 29-11

## Dynamic Calc property, 5-21

- and sparse dimensions, 5-23, 10-14
- applying, 29-7, 29-8, 29-18
  - in Application Manager, 29-22
- description, 9-20
- removing, 29-23

## dynamic calculations, 29-1

- asymmetric data, 29-18
- Attribute Calculations dimension, 10-31
- building, 29-8, 29-9
  - restrictions, 29-5, 29-10, 29-20
- building dimensions with, 29-23
- calc scripts and, 29-8, 29-21
- choosing members, 29-13
  - guidelines for, 29-13, 29-15
- choosing values, 29-6, 29-15
  - guidelines for, 29-7, 29-8
- database design, 7-14
- defined, 29-2
- defining order, 29-16, 29-17
- exporting data and, 29-21
- formulas applied to members, 26-6
- optimizing retrieval, 29-12
- partitioned databases, 29-25
- replicated partitions, 6-13
- retrieving values, 29-4, 29-10, 29-16, 29-21

- dynamic calculations (*Continued*)
    - transparent partitions, 6-22
    - usage overview, 29-5, 29-15
    - viewing log file contents, 29-10
  - dynamic dimension building
    - adding members to outlines, 13-13, 13-15, 13-17
    - generation references, 13-6
    - level references, 13-8
  - Dynamic Dimensionality, 1-3
  - dynamic dimensions
    - level references, 13-43
  - dynamic references
    - defining header information, 21-13
    - description, 14-23
    - record numbers in, 21-14
  - Dynamic Time Series Member Information dialog
    - box
    - defining aliases, 30-13
    - enabling/disabling members, 30-10, 30-11
  - Dynamic Time Series members, 30-7, 36-31
    - enabling/disabling, 30-9, 30-11
    - generation names for, 30-14
    - in shared areas, 30-16
    - inserting in report scripts, 36-31
    - listed, 30-8
    - reporting on, 38-8
    - selecting, 36-31
    - specifying aliases for, 30-13
- E**
- E, expense property code in data source, 14-25
  - East database
    - partitioning, 6-10
    - provided with Hyperion Essbase, xlv
  - Edit Alias dialog box, 11-6
  - Edit User dialog box, 17-5, 17-12
    - modifying access settings, 17-23
  - editing
    - See also* changing
    - aliases, 11-6
    - filters, 18-11
    - linked files, 12-8
    - partitions, 16-57
  - editing (*Continued*)
    - report scripts, 35-11, 35-17, 35-18, 35-20
    - security profiles, 17-12
    - user groups, 17-14
  - editors
    - calc scripts and, 31-10
    - ESSCMD commands and, 44-10
    - formulas and, 26-11
    - report scripts and, 35-11
  - electronic mailboxes, 12-2
  - ELSE operator
    - usage examples, 27-4, 27-5
  - ELSE statement, 26-29
  - ELSEIF statements
    - calc scripts, 31-17
    - formulas, 26-7, 26-29
  - empty database cells, 4-7
    - caution for storing, 4-16
  - empty fields
    - adding values to, 22-15
    - blank fields in rules file, 20-17
    - in data source, 20-16
  - emptying. *See* clearing
  - ENDARCHIVE command, 48-2, 48-3, 48-8
    - partitioned applications and, 6-19
  - ENDFIX command, 31-12, 31-17, 31-55
    - allocating costs, 32-8, 32-13
    - allocating values, 32-10
    - calculating product/market shares, 32-6
    - calculating range of values, 31-55
    - calculating subsets of values, 32-3
    - clearing data blocks, 31-53
    - clearing subsets of values, 31-53
    - copying subsets of values, 31-54
    - currency conversions and, 43-27
    - Intelligent Calculation and, 34-10, 34-14, 34-16
    - optimizing calculations with, 33-10, 33-11
  - ENDHEADING command
    - entering in report scripts, 36-18
    - modifying headings, 36-6
    - usage example, 37-25

- ENDIF statement
  - calc scripts
    - interdependent formulas in, 31-38
    - member formulas in, 31-37
    - semicolons with, 31-17
    - syntax, 31-16
  - formulas
    - purpose of, 26-29
    - semicolons with, 26-7
    - syntax, 26-7
  - usage examples, 27-4, 27-5
- ENDLOOP command, 31-12, 32-17
- end-of-file markers, 24-6
- end-of-line character, 26-6, 26-7
  - ESSCMD commands, 44-2
  - inserting in calc scripts, 31-15, 31-17, 31-40
  - inserting in formulas, 26-16
- enterprise analysis applications, 1-4
- Environment Variables list box, 46-5
- environments, 1-3
  - See also* UNIX platforms; Windows platforms
- equal signs (=)
  - in application and database names, 7-12
  - in calc scripts, 31-39
  - in dimension and member names, 8-15
  - in formulas, 26-15
  - in names in scripts and formulas, 8-16
  - in report scripts, 36-3
- equations, 26-28, 31-2
  - See also* formulas
  - cross-dimensional operator and, 33-10, 33-11
  - inserting in calc scripts, 26-28, 31-37
  - report scripts, 36-21
- .ERR files. *See* error log files
- error codes and numbers, 46-41
- error handling (Hyperion Essbase kernel), 40-7
- error log files, 46-26
  - default directory for, 24-1
  - empty, 24-2
  - from partial loads, 23-15
  - loading, 24-5
  - maximum number of records in, 23-18, 24-2
  - missing, 24-1
- error log files (*Continued*)
  - names and locations, 46-27
  - renaming, 24-5
  - resetting
    - record count, 24-2
    - setting, 23-12
- error message categories, 46-41
- error-handling commands, 44-11
- errors
  - calc scripts
    - checking for, 31-48
  - currency conversions, 43-18
  - ESSCMD, 44-11, 44-16
  - fixing for dynamic builds, 14-30
  - formulas and, 26-27
  - formulas and dynamically calculated members, 29-8
  - formulas, checking for, 26-25
  - handling fatal, 40-7
  - loading data, 23-10, 23-12
  - locking, 46-40, 47-14
  - matching member omitted, 38-8
  - mismatches, 42-13
  - partition validation, 16-39
  - pinpointing, 46-26
  - Report Extractor, 38-8
  - restructuring and, 40-17
  - storage allocation, 40-4
  - time-out, 24-2, 24-5
  - unknown member, 36-40
- ESBBEGINREPORT function, 36-52
- ESBENDREPORT function, 36-52
- ESBREPORT function, 36-52
- ESBREPORTFILE function, 36-52
- .ESM files, 40-16, 40-17, 42-11, 47-2
- .ESN files, 40-16, 47-2
- .ESR files, 47-2
- ESSBASE, 45-1, 45-4
- Essbase
  - architecture, 1-4, 4-1
  - development features described, 1-2
  - displaying software version number, 45-5, 45-12
  - guide prerequisites, xliii

*Essbase (Continued)*

- products summarized, 1-1
- program files, 47-1
- quick reference to menus and dialog boxes, 46-18
- quitting, 45-5, 45-12
- terms described, 3-5
- Essbase client. *See* clients
- ESSBASE command, 45-2
- Essbase Config page (Server Information), 46-5
- Essbase Server Agent. *See* Server Agent
- Essbase server. *See* server
- Essbase System Login dialog box, 23-4
- ESSBASE.BAK, 17-2
- ESSBASE.CFG, 38-5, 46-5
- ESSBASE.LOG, 45-4, 45-16, 46-31
- ESSBASE.SEC, 17-2
  - and filters, 18-2
  - backup of, 17-2
- ESSBEGINREPORT function, 36-52
- ESSCMD, 45-13
  - See also* specific command
  - canceling operations, 44-3, 44-8
  - caution for processing commands, 44-8
  - changing Hyperion Essbase kernel settings, 41-3, 41-7
  - checking structural integrity, 42-13
  - command reference, xliv
  - commands and case-sensitivity, 44-5
  - creating report scripts, 35-5
  - defined, 44-1
  - enabling Cache Memory Locking, 41-13
  - entering commands, 44-8
  - error-handling, 44-11, 44-16
  - loading update log files, 48-8
  - operational modes, 44-1
  - performance-related commands, A-2
  - referencing files, 44-3
  - replicating data, 6-14
  - running, 44-3, 44-5
    - calc scripts, 31-24, 31-31, 31-34
    - caution for pausing systems and, 44-8
    - in interactive mode, 44-2
    - reports, 35-27

*ESSCMD (Continued)*

- setting
  - cache size, 41-9, 41-12
  - index page size, 41-15
  - isolation levels, 41-7
  - substitution variables, 36-33
  - transaction isolation levels, 41-18
- shutting down server, 45-3
- specifying
  - data compression, 41-28
  - disk volumes, 41-24
- starting, 44-6
  - prerequisites, 44-5
- syntax, 44-2
- updating replicated partitions, 16-62
- ESSCMD script files
  - creating, 44-1, 44-10
  - naming, 44-9
  - quotation marks in, 44-2
  - running, 44-10
  - sample, 44-12
  - within operating system batch files, 44-15
- ESSCMD.EXE, 44-6
- ESSENDREPORT function, 36-52
- ESSGBEGINREPORT function, 36-52
- ESSGREPORTFILE function, 36-52
- Essnet library, 45-13
- ESSREPORT function, 36-52
- ESSREPORTFILE function, 36-52
- ESSSVR (application server), 45-13, 45-14
- estimating disk space, 15-3, 15-12
  - linked reporting objects, 15-13
  - partitioned applications, 15-12
- Euro currency symbol, 20-4
- EUROPEAN command
  - overriding, 36-23
  - usage, 36-27
- event log files, 29-11, 46-28
  - calc scripts and, 31-9
  - deleting, 46-29, 46-32
  - server-specific, 46-31
  - viewing contents, 29-10, 31-9, 46-28
- events, 45-14
- Excel spreadsheets. *See* Spreadsheet Add-in
- exception error logs, 46-26
  - names and locations, 46-27

- Exception Handler, 46-26
    - location of writes, 46-27
  - ExceptionLogOverwrite parameter, 46-27
  - exchange rates, 43-2, 43-22
    - calculating options, 43-30
    - defining, 9-10
    - sample database, 43-2
  - exclamation points (!)
    - See also* bang command
    - in names in scripts and formulas, 8-16
  - excluding members from consolidations, 9-18
  - exclusive locks. *See* Write locks
  - .EXE files, 47-4
  - Execute Calc Scripts dialog box, 17-26
  - Existing Members options in dimension build, 14-10
  - EXIT command, 44-6, 45-3, 45-5
    - usage overview, 45-12
  - exiting
    - See also* closing; quitting; stopping
    - Essbase, 45-5, 45-12
    - Partition Wizard, 16-41
  - expanded blocks, 15-8
  - expanding
    - alias member lists, 11-6
    - Calc Script Editor lists, 31-45, 31-47, 31-48
    - Formula Editor lists, 26-21, 26-24, 26-25
    - outline hierarchies, 8-3
  - expense property, 5-31, 9-9
    - specifying in data source, 14-25
  - expense vs. non-expense items, 26-39
  - exponentiation, 26-38
  - EXPORT command, 48-4
    - partitioned applications and, 6-19
  - export files, 8-12, 23-2
  - Export Server Alias Table Object dialog box, 11-13
  - Export Server Outline Object dialog box, 8-11
  - EXPORT utility, 48-3, 48-4
  - exported data reloads, 48-4
  - exporting
    - alias tables, 11-11, 11-13
    - data, 29-21, 39-1, 39-13
      - for backups, 48-3
      - into other forms, 36-15
      - reporting example, 37-37
    - outlines, 8-10, 8-11
  - external data sources
    - for loading data, 5-17, 20-3, 20-8, 23-1
    - linking to cells, 12-2
    - prerequisites for loading, 23-2
    - supported, 23-2
  - extraction
    - characters ignored, 36-3
    - commands, 35-9, 36-27
    - data, 38-7, 39-1
  - extraction commands
    - defined, 36-1
- ## F
- F, first time balance code in data source, 14-25
  - @FACTORIAL function, 26-38
  - factorials, 26-38
  - failures
    - calculations, 17-37
    - getting information about, 46-27
    - preventing, 6-8, 6-12
    - recovering from, 24-5, 48-5, 48-6
      - suggested procedures, 48-7
    - restructuring and, 40-17
    - transaction rollbacks and, 42-11
  - fatal errors, 40-7
  - FEEDON command, 36-10
  - fetches, 40-2
  - Field Edits page (Data File Properties), 22-11
  - field names
    - adding prefixes or suffixes to, 22-20
    - assigning to data fields, 20-16
    - changing case, 22-16
    - SQL data sources, 22-12
    - validating, 21-23
  - field operations, 20-10
  - Field Properties dialog box
    - adding members, 13-13, 13-15, 13-17
    - adding prefixes and suffixes, 22-20
    - applying case conversions, 22-17
    - converting white space to underscores, 22-19
    - defining columns as fields, 22-21
    - mapping fields, 22-3
    - naming fields, 22-13

Field Properties dialog box (*Continued*)

- removing white space, 22-18
- replacing text strings, 22-14
- scaling data values, 22-26
- setting field types, 14-13

## field types

- data sources, 20-2, 20-3, 20-5
  - invalid, 20-6
- dimension building properties, 14-15
- dynamically referencing, 14-23
- restrictions for rules files, 14-17
- setting for record selection, 21-16, 21-17
- setting member properties, 14-13, 14-25
- shared members, 13-33, 13-36, 13-38, 13-39, 13-40, 13-42
  - caution for creating, 13-41
- specifying, 13-5, 14-12
- valid build methods, 14-15

## fields

- See also* columns
- adding prefixes or suffixes to values, 22-20
- arranging, 14-17, 22-5, 22-11
- building dynamically, 13-7, 13-10
- changing case, 22-16
- clearing existing values, 22-24
- concatenating multiple, 22-6
- copying, in rules files, 22-7
- creating, 22-10, 24-12
  - by splitting, 22-9
  - with joins, 22-6, 22-7
- defined, 20-1
- defining columns as, 22-21
- defining operations for rules files, 13-2
- delimiters/separators. *See* file delimiters
- displaying in Data Prep Editor, 21-4
- empty
  - adding values to, 22-15
  - in data source, 20-16
  - in rules file, 20-17
  - replacing with text, 22-15
- entering data, 20-4, 20-5
- excluding specific from data loads, 20-17
- formatting rules, 20-12, 20-14, 20-15
- invalid, 20-6

fields (*Continued*)

- manipulating in rules files, 22-1
  - mapping
    - changing case, 22-16
    - inserting text in empty fields, 22-15
    - replacing text strings, 22-14
    - specific only, 22-2, 22-3, 22-4
    - to member names, 22-12
  - moving, 22-5
  - naming, 22-12
  - position in rules file, 13-22, 14-17
  - processing nulls in, 13-7, 13-10
  - removing white space in, 22-18
  - reversing values in, 22-27
  - select/reject criteria, 14-20
  - selecting, 22-2
  - setting retrieval specifications, 21-16, 21-17
  - size and optimal loads, 24-12
  - undoing operations on, 22-11
- File Delimiter page (Data File Properties), 21-10
- file delimiters
- customizing, 21-11
  - formatting rules, 20-18
  - in header information, 14-24
  - mapping files, 16-20
  - setting, 20-6, 21-10
- file locks, 16-3, 46-40
- file system types, 46-7
- file-name extensions
- batch files, 44-9
  - calc scripts, 31-24
  - error log files, 46-27
  - ESSCMD script files, 44-9
  - in ESSCMD, 44-4
  - index files, 40-16
  - listed, 47-1
  - live reporting object files, 40-5
  - outline change files, 46-35, 46-39
  - outline files, 40-16
  - report files, 35-25
  - report scripts, 35-13
  - restructuring and, 40-16
  - rules files, 21-25
  - temporary files, 40-16
  - text files, 21-6
  - trace files, 46-33

## files, 47-5

*See also* data files; rules files; text files  
 attaching external to cells, 12-1, 12-2  
 backing up, 48-2, 48-3  
 caution for recovery and, 42-11  
 compatibility across platforms, 47-15  
 creating trace, 46-33, 46-35  
 editing linked, 12-8  
 estimating size, 15-5  
 implementing security for, 17-29  
 importing, 23-2, 44-13, 44-14  
 loading, 15-3  
 locating end-of-file markers, 24-6  
 location of sample report scripts, 37-1  
 logging data load errors, 23-12  
 logging dimension builds, 14-30  
 logging outline changes, 16-48  
 opening, 21-5, 21-7, 21-8  
 program, 47-1  
 referencing in ESSCMD, 44-3  
 renaming with FTP, 47-18  
 restrictions for linking, 12-3  
 restructuring, 40-15  
 running multiple report scripts, 35-27  
 selecting multiple for loading, 23-7, 23-8  
 sending report output to, 35-24  
 setting maximum size, 41-20  
 specifying size for linked, 12-4, 12-5  
 storing, 12-3  
 temporary for full restructures, 40-16  
 transferring compatible, 47-18  
 types defined, 7-2  
 types described, 47-1  
 updating log, 16-53  
 with long names, 47-18

Files page (Database Information), 46-16

Filter Access privilege, 17-25

Filter privilege, 17-26

filters, 18-2

and attribute functions, 18-10  
 AND relationship, 18-9  
 assigning, 18-15  
 copying, 18-12

filters (*Continued*)

creating, 6-31, 18-3, 18-6  
 for partitioned databases, 6-30  
 defined on separate rows, 18-8  
 defining in Application Manager, 18-3, 18-5  
 deleting, 18-14  
 editing, 18-11  
 in the security (.SEC) file, 18-2  
 inheriting, 18-2, 18-15, 18-16  
 on entire members, 18-8  
 on member combinations, 18-9  
 OR relationship, 18-8  
 overlap conflicts, 18-16, 18-17  
 overriding, 16-5, 25-12  
 overview, 18-1, 18-2  
 removing, 18-14  
 renaming, 18-13  
 restrictions on applying, 18-15  
 saving, 18-2  
 using member set functions, 18-10  
 viewing existing, 18-3  
 write access to database, 13-46

Filters dialog box, 18-4

financial applications, 3-12

allocating costs, 32-7  
 allocating values, 32-9, 32-13  
 calculating product and market share, 32-6  
 comparing actual to budgeted expense, 9-9, 26-8  
 containing time-sensitive data, 5-16  
 estimating sales and profits, 32-17  
 example for building, 5-3  
 forecasting values, 32-21  
 getting budgeted data values, 32-3  
 getting variance for actual to budgeted, 26-39,  
 32-2

loading new budget values, 32-5

financial functions, 26-4, 26-45

formulas and, 31-50, 33-6, 33-7  
 Intelligent Calculation and, 34-19

Find dialog box

Calc Script Editor, 31-42, 31-46  
 Formula Editor, 26-19, 26-22  
 Report Editor, 35-17

- finding
  - end-of-file markers, 24-6
  - members in Calc Script Editor, 31-46
  - members in Formula Editor, 26-22
  - specific values, 4-6, 26-44
  - text in formulas, 26-19
  - text in report scripts, 35-17, 35-18
- first time balance property, 9-5
  - specifying in data source, 14-25
- first-time calculations, 34-6
- first-time users, 2-1
- FIX/ENDFIX command, 31-12, 31-17, 31-55
  - allocating costs, 32-8
  - allocating values, 32-10, 32-13
  - calculating product/market shares, 32-6
  - calculating range of values, 31-55
  - calculating subsets of values, 32-3
  - clearing data blocks, 31-53
  - clearing subsets of values, 31-53
  - copying subsets of values, 31-54
  - currency conversions and, 43-27
  - Intelligent Calculation and, 34-10, 34-14, 34-16
  - optimizing calculations with, 33-10, 33-11
- FIXCOLUMNS command, 37-21
- fixed-size overhead, 15-8, 40-8
- fixed-width columns, 21-11
- flags (partitions), 6-6
- flat dimensions, 33-4, 33-22
- flat files, 15-13
  - loading, 23-2
- FLAT2SQL.REP, 37-38
- flipping data values, 22-27
- Font dialog box, 8-26, 35-23
- fonts, 8-26, 35-23
- forecasting values
  - examples, 32-21
  - functions, 26-5, 26-42
- foreground startup, 45-2
- formats
  - See also* formatting commands
  - alias tables, 11-3
  - calculated data, 36-16, 36-20
  - columns for data load, 20-22, 20-23
  - comments, 9-27
  - formats (*Continued*)
    - data sources, 20-12, 20-14, 20-15, 20-17, 20-18
      - overview, 20-2
    - export, 36-15
    - free-form data sources, 20-18, 20-19, 20-22
    - header information, 14-23
    - import specifications and, 39-1
    - linked object restrictions, 12-3
    - member mappings, 16-19
    - report output, 35-7
    - sample report, 37-2, 37-17, 37-35, 37-37
    - suppressing in reports, 36-14, 36-23
    - tab-delimited, 36-15
  - formatting commands, 35-29, 36-8
    - calculations, 36-16, 36-20
    - caution for usage, 36-48, 36-51
    - defined, 36-1
    - display options, 36-22, 36-26
    - listed, 36-9
    - nesting, 36-2
    - output, 36-15
    - report headings, 36-10, 36-13, 36-14, 36-15
  - formatting styles (documentation), xlv
- Formula Editor
  - building formulas, 26-11, 26-13, 26-15, 26-20
  - checking syntax, 26-25, 26-26
  - deleting formulas, 26-13, 26-14
  - expanding/contracting dimensions, 26-24
  - expanding/contracting member branches, 26-21
  - finding and replacing text, 26-19
  - opening, 9-29, 26-12
  - saving formulas, 26-14
  - searching for members, 26-22
  - undoing last action, 26-15
- formula field type
  - in header records, 14-24
  - in rules files, 14-15
  - nulls and, 13-7, 13-10
- FORMULA. *See* formula field type
- formulas, 26-1, 31-2, 33-5
  - adding aliases, 26-25
  - adding text to, 26-15
  - adding to calc scripts, 31-16, 31-36, 31-38, 31-51
  - adding to report scripts, 37-32

formulas (*Continued*)

- allowing changes, 14-11
- applied to members, 26-5, 26-33, 26-35, 26-36, 28-6, 31-11
- applying to data subsets, 31-54
- associating with dimensions or members, 9-28
- calculating twice, 33-24, 33-31, 33-34
  - with Intelligent Calculation, 33-32, 33-33
- caution for sparse dimensions, 31-50
- changing, 26-13
- complex, 33-7, 33-12
- creating, 5-31
  - examples, 26-8, 26-28, 26-31, 27-1
  - syntax for, 26-6
  - with Formula Editor, 26-13
  - writing equations, 26-27
- cross-dimensional references in, 33-7, 33-12
- defined, 26-1
- deleting, 26-14
- displaying, 8-25, 26-13
- dynamic calculations and, 29-7, 29-8
- errors in, 26-25, 26-27
- examples, 27-6
- finding and replacing text in, 26-19
- grouping in calc scripts, 31-50, 31-51, 33-13
- in partitioned applications, 6-21, 6-22
- in time and accounts dimensions, 30-6
- inserting operators, 26-15
- inserting variables, 26-48
- Intelligent Calculation and, 34-18
- names with special characters, 8-16, 8-17
- nesting, 31-51
- optimizing, 33-13
- partitions and, 26-49
- performance considerations, 33-5, 33-8
- printing, 26-14
- saving, 26-14
- setting conditions for, 26-29, 26-31
- shared members and, 9-23
- simple, 33-6, 33-13
- top-down, 33-12
- troubleshooting, 26-25, 26-27
- types described, 26-27
- forward calculation references, 28-8
- FoxPro databases. *See* SQL databases
- fragmentation (defined), 40-14

- fragmentation allowance, 15-10
- free-form
  - data load, 20-18
  - data sources, 20-18
    - formatting, 20-19, 20-22
    - optimizing, 24-11
  - reports, 35-28
- freeing memory, 46-6
- FTP file transfers, 47-18
- full restructure, 40-15
  - temporary files created, 40-16
- function syntax, xlirii
- Function Templates dialog box, 26-17
- functions, 1-5, 5-31, 26-4
  - See also specific @function*
  - allocation, 26-42
  - API, 12-3
  - applying in filter definitions, 18-6
  - applying to subsets, 31-55
  - Boolean, 26-29
  - calc scripts and, 31-40
  - date and time, 26-46
  - defining multiple members and, 26-35
  - defining partitioned areas with, 16-9, 16-23, 16-34
  - financial calculations with, 26-45
  - forecasting, 26-42
  - formulas and, 26-4, 26-17, 33-6, 33-7
  - generating member lists, 10-37, 26-36
  - Intelligent Calculation and, 34-19
  - mathematical, 26-38
  - member set, 26-36
  - range, 26-43
  - relationship, 26-44
  - report generation, 36-52
  - returning specific values, 26-44
  - statistical, 26-41
- fundamentals, 2-2
- future values, calculating, 26-43

**G**

- @GEN function, 26-44
- GEN field type. *See* generation field type
- GEN numbers. *See* generation reference numbers
- General page (Database Information), 46-10

- General page (Database Settings), 41-6
- Generate Currency Outline button, 43-18
- Generate Currency Outline dialog box, 43-19
- generating
  - account reporting values, 30-1
  - member lists, 26-4, 26-36, 31-55
  - reports, 36-1, 36-7, 38-6
    - basic techniques, 35-2, 35-11
    - free-form, 35-28
    - from the command line, 35-5
    - with API function calls, 36-52
- Generation and Level Names dialog box, 8-23
- generation field type
  - arranging in rules files, 14-17
  - in header records, 14-24
  - in rules files, 14-15
  - null values and, 13-7
- generation names, 3-8
  - adding to report scripts, 36-29, 36-38
  - creating, 8-23, 14-8
  - displaying, 8-24
  - for Dynamic Time Series members, 30-14
  - static, 36-40
- generation reference numbers
  - associating attributes dynamically, 13-20
  - building shared members, 13-33
  - defined, 13-6
  - Dynamic Time Series members, 30-7, 30-14
  - entering in rules files, 14-16
  - generating, 26-44
- generation references build method
  - creating shared members, 13-33
  - discussion, 13-5
  - guidelines for using, 13-4
  - null processing, 13-7
  - sample rules file, 13-6, 13-34
  - shared members, 13-33
  - valid build types, 14-15
- generation references defined, 13-5
- Generation/Level Names page (Dimension Properties), 14-8
- generations
  - checking for, 26-30
  - defined, 3-7, 13-5
  - duplicate, 13-33
  - generations (*Continued*)
    - Dynamic Time Series members and, 30-7, 30-9, 30-14
    - levels vs., 3-7
    - naming, 3-8, 8-23, 14-8
    - reversing numerical ordering, 3-7
    - sharing members, 13-33, 13-37
      - at multiple, 13-32
    - sorting on, 36-44
  - generic passwords, 17-10
  - @GENMBRS function, 26-37
  - GENREF.RUL, 13-6
  - GENREF.TXT, 13-6
  - GETALLREPLCELLS command, 16-62
  - GETAPPINFO command, 46-8
  - GETAPPSTATE command, 46-3
  - GETATTRIBUTESPECS command, 10-16
  - GETATTRINFO command, 10-18
  - GETCRDBINFO command, 46-12
  - GETDBINFO command, 46-9, 46-15
  - GETDBSTATE command, 46-11
  - GETDBSTATS command, 40-11, 46-14
  - GETMBRCALC command, 26-13
  - GETPARTITIONOTLCHANGES command, 16-56
  - getting started with Hyperion Essbase, 2-1, 2-2
  - GETUPDATEDREPLCELLS command, 16-62
  - GETVERSION command, 45-12
  - global
    - formatting commands, 36-8, 36-46
  - Global Access layer (security and privileges), 17-29
  - global access settings
    - defining, 17-31, 17-34, 17-35
    - minimum database access, 17-32
    - overriding, 17-33
    - types listed, 17-30
  - global calculation commands, 31-13
  - Global ESSBASE.CFG Settings list box, 46-5
  - global options (dynamic builds), 14-20
  - global placeholders, 7-14
  - Global Properties page (Field Properties)
    - adding prefixes and suffixes, 22-20
    - applying case conversions, 22-17
    - converting white space to underscores, 22-19
    - defining columns as fields, 22-21

Global Properties page (*Continued*)  
 removing white space, 22-18  
 replacing text strings, 22-14  
 scaling data values, 22-26  
 setting ignored values, 22-3

global replacement  
 data load, 22-15

Global Setting page (Dimension Build Settings),  
 14-20

goal seeking calculations, 32-17

GOTO command, 44-11

granularity, 6-9

graphs, 12-1

greater than signs (>)  
 in application and database names, 7-12  
 in names in scripts and formulas, 8-17

gridlines (Data Prep Editor), 21-4

Group Membership dialog box, 17-11

groups  
 assigning filters to, 18-15  
 assigning privileges to, 17-5  
 assigning users to, 17-11, 17-15  
 copying security profiles, 17-18  
 creating member, 9-22, 24-8  
 creating user, 17-11, 17-14, 17-16  
 defined, 17-4, 17-14  
 defining security, 17-7, 17-9, 17-16  
 deleting, 17-20  
 deleting users, 17-11  
 editing user information, 17-14  
 formulas and dimensions in calc scripts, 31-50,  
 31-51, 33-13  
 getting list of, 17-8  
 modifying access settings, 17-12, 17-22, 17-27,  
 17-28  
 overriding column, 36-8  
 renaming, 17-21  
 returning valid users, 17-11, 17-16  
 security settings, 17-4  
 security types, 17-5  
 selecting members for report, 36-7  
 viewing current settings, 17-27, 17-28

@GROWTH function, 26-45

guest accounts, 6-26

guidelines  
 analyzing database design, 5-10

## H

.H files, 47-5

handles, 12-3

handling missing values, 36-25  
 reporting example, 37-4

handling zero values, 36-25

hardware platforms, 1-3

hash tables, 33-4, 33-22

Header Definition page (Data Load Settings), 21-12

header information  
 adding, 14-23, 14-24, 21-11, 21-13  
 dynamic references, 21-11  
 formatting, 14-23

header records  
 creating, 21-11, 21-13  
 defined, 21-11  
 defining load rules, 14-23  
 identifying missing dimensions with, 20-13  
 skipping, 20-11, 21-14  
 specifying location of, 14-24

Header Records page (Data File Properties), 21-14

headers. *See* header information; header records

HEADING command, 36-6

HEADING1.REP, 37-26

HEADING2.REP, 37-32

headings  
*See also* column headings; field names; row  
 headings  
 adding to complex reports, 36-4, 36-17  
 adding to free-form reports, 35-29  
 currency conversion, 36-51  
 customizing, 36-6, 37-23  
 display options, 36-6, 36-14  
 forcing immediate display, 36-6  
 formatting, 36-10, 36-13, 36-14, 36-15  
 multiple layout reports, 37-36  
 page, 35-7  
 suppressing display, 36-14  
 using attributes in, 36-5

help  
 API functions, 36-52  
 calc scripts, 31-12  
 configuration variables, 38-2  
 configurations, 38-5

- help (*Continued*)
    - files, 47-4
    - loading sample applications, xliv
    - Report Writer commands and definitions, 35-5
    - Server Agent, 45-5
  - Help button (Application Desktop), 7-8
  - HELP command, 45-5
  - hierarchies
    - Calc Script Editor
      - aliases, 31-48
      - dimensions, 31-47
      - members, 31-45
    - calculation order and, 28-3
    - data, 3-4
      - relationships defined, 3-5, 3-6
    - dynamic builds, 13-3
    - Formula Editor
      - aliases, 26-25
      - dimensions, 26-24
      - members, 26-21
    - outlines, 8-1
      - expanding/contracting, 8-3
      - unbalanced in outlines, 36-37
  - history-to-date calculations, 30-8
  - .HLP files, 47-4
  - hosts, 16-63
  - HP-UX servers. *See* UNIX platforms
  - H-T-D time series member, 30-8, 30-14
  - HTML files, 12-2
  - Hyperion Essbase
    - fundamentals, 2-2
    - getting started tips, 2-1, 2-2
  - Hyperion Essbase kernel
    - active transactions and failures, 42-11
    - allocating storage, 40-11
    - changing settings, 41-2, 41-3
      - scope and precedence, 41-4
      - with Application Manager, 41-5
      - with ESSCMD, 41-3, 41-7
    - components, 40-2
    - compressing data, 40-8, 40-10, 40-11
    - cross-platform compatibility, 47-16
    - customizing, 41-2, 41-3
    - error handling, 40-7
  - Hyperion Essbase kernel (*Continued*)
    - linked reporting objects and, 15-13
    - locking procedure, 42-5
    - overview, 40-1, 41-2
    - setting cache size, 41-8, 41-9
    - setting index page size, 41-14, 41-15
    - setting isolation levels, 41-15, 41-16, 41-19
    - specifying data compression, 41-27, 41-29
    - specifying disk volumes, 41-20, 41-26
    - starting, 40-7
  - hyphens (–)
    - in dimension and member names, 8-15
    - in member names, 20-5
    - in names in scripts and formulas, 8-16
    - in report scripts, 36-3
- I**
- @IALLANCESTORS function, 26-36
  - @IANCESTORS function, 26-36
  - IANCESTORS command
    - usage, 36-28
  - IBM DB2 relational storage manager, 40-20
  - @ICHILDREN function, 26-36
  - ICHILDREN command
    - usage, 36-28
    - usage example, 37-7
  - .ICO files, 47-4
  - identical values, 5-28
  - identifying data sources, 5-4
  - @IDESCENDANTS function, 26-36, 31-55
  - IDESCENDANTS command
    - usage, 36-28
    - usage example, 37-3
  - IF/ELSEIF... statements
    - calc scripts
      - nested statements, 31-16
    - formulas
      - nested statements, 26-7
      - purpose of, 26-29
  - IF/ENDIF statements
    - calc scripts
      - interdependent formulas in, 31-38
      - member formulas in, 31-37
      - semicolons with, 31-17
      - syntax, 31-16

IF/ENDIF statements (*Continued*)

- formulas
  - purpose of, 26-29
  - semicolons with, 26-7
  - syntax, 26-7
- usage examples, 27-4, 27-5

## IFERROR command, 44-11

## Ignore Conflicts option, 14-10

## Ignore Tokens page (Data File Properties), 22-4

## ignoring

- #MISSING and zero values, 9-7, 30-5
- end-of-file markers, 24-6
- fields when mapping, 22-2, 22-3, 22-4
- lines in rules files, 20-11, 21-14
- multiple fields in data load, 21-18
- specific fields in data load, 21-17
- specific records in data load, 21-16

## @ILSIBLINGS function, 26-37

## immediate restructuring, 40-19, 40-22

## IMMHEADING command, 36-6

## implementing security measures, 25-12

- for users and groups, 17-5, 17-7, 17-9, 17-16
- globally, 17-29, 17-31, 17-34, 17-35
- guidelines for, 2-3
- partitioned databases, 6-30, 6-31, 6-32
- planning, 5-17
- system server, 17-36, 17-37, 17-39

## implied shared relationships, 5-22, 9-23, 24-3

## Import Member Mappings dialog box, 16-19

## Import Server Alias Table Object dialog box, 11-12

## Import Server Outline Object dialog box, 8-13

## importing

- alias tables, 11-11
- data, 39-1, 39-16
- files, 23-2, 44-13, 44-14
- member mappings, 16-19
- outlines, 8-10, 8-13

importing database files *See* reloading database files

## improper shutdown, 17-33

## improving performance, 6-13, 6-20, 6-25

*See also* optimizing

## inaccessible applications, 17-33

## inactive users, 17-41

## INCBUILDDIM command, 14-2

## INCEMPTYROWS command

- overriding, 36-14
- usage, 36-23

## INCFORMATS command, 36-23

## INCMASK command, 36-23

## INCMISSINGROWS command, overriding, 36-14

## inconsistent values, 40-8

## incorrect data values, 24-2

## incremental data load, 33-3

## incremental restructuring, 40-19

- disabled, 40-20
- files, 47-2

## INCRESTRUC parameter, 40-19

- linked reporting objects and, 40-20

## INCZEROROWS command, 36-23

- overriding, 36-14

## .IND files, 40-16, 47-2

## INDENT command, 36-24

## indentation

- turning off, 37-20
- usage, 36-24

## INDENTGEN command

- usage, 36-24
- usage example, 37-31

## index

- advantage of small, 4-13
- checking structural integrity, 42-13
- defined, 4-4
- determining optimal size, 4-11, 4-12, 4-13, 33-2
- disadvantage of large, 4-11
- dynamic calculations and, 29-24
- managing, 40-3, 40-4
- optimizing, 33-32, 33-33
- process for retrieval, 15-3
- rebuilding. *See* restructuring
- removing references to linked files, 12-8
- restructuring, 40-17
- retrieving linked reporting objects, 12-3
- spanning multiple pages, 40-3
- updating, 40-15
- usage described, 4-5

## index cache

- as storage unit, 15-2
- determining size, 15-15
- managing, 40-3

- index cache (*Continued*)
  - optimizing read/writes, 24-10
  - setting size, 15-16, 33-14
    - with Application Manager, 41-8
    - with ESSCMD, 41-9
    - with Hyperion Essbase kernel, 41-8
  - updating, 41-9
- index entries, 4-4, 4-5
  - managing, 40-3
- index files, 40-16
  - associated with current application, 46-16
  - caution for recovery and, 42-11
  - creating, 40-12
  - cross-platform compatibility, 47-16
  - defined, 15-2
  - size considerations, 15-3
  - storing, 40-12, 41-20
- Index Manager, 40-2
  - overview, 40-3
- index page
  - linked reporting objects and, 40-5
  - managing, 40-3
  - setting size, 41-14, 41-15
  - updating, 41-15
- index page files, 40-16
- index searches, 4-6
  - viewing information about, 46-14
- information dialog boxes, 46-2
- information flow, 5-4
- informational messages, 33-4, 33-23
- inheritance
  - currency categories, 43-9
  - filters, 18-2, 18-15, 18-16
  - privileges, 17-29, 17-30
  - properties, 9-2
- initialization process (Hyperion Essbase kernel), 40-7
- initializing rules files, 14-31
- .INM files, 40-16, 40-17
- .INN files, 40-16, 47-2
- input blocks, 28-3, 48-4
- input data, 5-25, 25-1, 48-4
  - restructuring and, 40-18, 40-28
- input/output (I/O), 40-2
- Insert Arguments option, 26-18, 31-41
- inserting. *See* adding
- installation information, 46-4
- installer
  - security level of, 17-5
- installing OLAP Server, xliii
- installing Personal Essbase, 39-1
- insufficient privileges, 17-29
- @INT function, 26-38
- integers. *See* numbers; values
- integrity. *See* data integrity
- Intelligent Calculation
  - block size and efficiency, 33-2
  - controlling with calc scripts, 31-50
  - currency conversions and, 34-20, 43-28
  - default calculations and, 33-31
  - enabling/disabling, 33-33, 34-5
  - large indexes and, 33-32
  - limitations, 34-4, 34-18
  - overview, 34-1, 34-2, 34-18
  - partial calculations and, 31-54
  - recalculating data and, 34-6
  - restructuring and, 40-15
  - setting as default, 34-5
  - time balance properties and, 30-2
  - turning on/turning off, 34-5
  - two-pass calculations and, 33-28, 33-32, 33-33
- Intelligent Calculation option, 28-14
- interactive mode (ESSCMD)
  - canceling operations, 44-8
  - command-line syntax for, 44-2
  - defined, 44-1
  - entering commands, 44-8
  - file-name extensions and, 44-4
  - overview, 44-6
  - running ESSCMD from, 44-2
- interdependent values, 26-33, 31-38
- interdimensional irrelevance, 5-15
- @INTEREST function, 26-45
- interest, calculating, 26-45
- interfaces, 1-5, 7-7
  - accessing linked objects and client, 12-4
  - customizing, 1-5
- international applications. *See* locales
- international formats, 36-23, 36-27

Interntl database, 43-1, 43-3
 

- changing Scenario dimension in, 43-14
- contents described, 43-3
- linking, 43-24
- opening outline, 43-8
- provided with Hyperion Essbase, xliv
- saving outline changes, 43-17

 interpolation, with spline, 26-42
   
invalid fields in data sources, 20-6
   
inventory, 26-33
 

- calculating first/last values, 30-3, 30-4
- calculating period-to-date values, 27-1
- currency categories and, 43-9
- getting averages, 9-6
- tracking, 5-3, 30-1

 investments, 26-46
   
IPARENT command
 

- usage, 36-28

 @IRR function, 26-45
   
irrelevant data, 5-15
   
@IRSIBLINGS function, 26-37
   
@ISACCTYPE function, 26-29
   
@ISANCEST function, 26-29
   
@ISCHILD function, 26-29
   
@ISDESC function, 26-30
   
@ISDESCENDANTS function, 22-25
   
@ISGEN function, 26-30
   
@ISIANCEST function, 26-29
   
@ISIBLINGS function, 26-37
   
@ISICHILD function, 26-30
   
@ISIDESC function, 26-30
   
@ISIPARENT function, 26-30
   
@ISISIBLING function, 26-30
   
@ISLEV function, 26-30
   
@ISMBR function, 26-30, 26-48
   
isolation levels, 15-4, 42-2
 

- calculations and, 33-23
- canceling calculations and, 25-11
- caution for data loads, 24-5
- committed and uncommitted access, 42-10
- locks and, 40-6, 42-6, 42-7, 42-9
- rollbacks and, 42-11, 42-12

isolation levels (*Continued*)
 

- setting
  - with Application Manager, 41-17
  - with ESSCMD, 41-7, 41-18
  - with Hyperion Essbase kernel, 41-15, 41-16
- updating, 41-19
- @ISPARENT function, 26-30
- @ISSAMEGEN function, 26-30
- @ISSAMELEV function, 26-30
- @ISSIBLING function, 26-30
- @ISUDA function, 26-30

## J

Join Fields dialog box, 22-6
   
joining fields, 20-17, 22-6

## K

kernel settings, options, 41-2

## L

L, last time balance code in data source, 14-25
   
label members
 

- See also* label only property
- dynamically calculating, 29-5
- storing, 15-4

 label only property, 5-22
 

- defining, 9-22
- description, 9-20
- specifying in data source, 14-25
- usage example, 5-29

 labels
 

- defining, 37-26
- replacing missing values with, 36-25, 37-4

 language support (programming), 1-5, 36-52
   
languages
 

- country-specific dimensions, 5-20, 9-11
- requirements for partitioning, 6-9

 large databases, 13-1, 33-4, 33-8
   
large-scale reports, 35-1, 38-7
   
last time balance property, 9-5
 

- example, 9-5
- specifying in data source, 14-25

- LATEST command, 36-32
- latest time period, 30-14
- layout commands, 36-3, 36-6
- layouts. *See* page layouts
- .LCK files, 47-4
- leading spaces, 22-18
- leaf members
  - defined, 3-6
  - restructuring, 40-28
- leaf nodes, 3-6
- less than signs (<)
  - in application and database names, 7-12
  - in dimension and member names, 8-15
  - in names in scripts and formulas, 8-17, 36-3
  - in report scripts, 35-9
- @LEV function, 26-44
- level 0 blocks, 28-3, 34-11
  - exporting, 48-4
  - restructuring, 40-18, 40-28
- level 0 members, 3-6
  - See also* leaf members
  - calculations on, 28-4, 29-5, 29-7
  - in two-dimensional reports, 39-14
  - required in source data, 14-17
  - storing, 28-3
- level field type
  - arranging in rules files, 14-17
  - in header records, 14-24
  - in rules files, 14-15
  - null values in, 13-10
  - sharing members, 13-38
- level names, 3-8
  - adding to report scripts, 36-29, 36-38
  - creating, 8-23, 14-8
  - displaying, 8-24
  - static, 36-40
- level numbers
  - determining for shared members, 13-35, 13-40
  - generating, 26-44
- level reference numbers
  - associating attributes dynamically, 13-20
  - entering in rules files, 14-16
- level references
  - described, 13-8
  - sample rules file for, 13-35
- level references build method, 13-8
  - creating multiple roll-ups, 13-43
  - creating shared members, 13-35, 13-38, 13-40
  - example rules file, 13-9, 13-43
  - guidelines for, 13-4
  - null processing, 13-10
  - sample rules file for, 13-35
  - shared members, 13-35, 13-38, 13-40
  - valid field types, 14-15
- LEVEL.RUL, 13-9
- LEVEL.TXT, 13-9
- LEVELMUL.RUL, 13-43
- levels
  - checking for, 26-30
  - defined, 3-7
  - naming, 3-8, 8-23, 14-8
  - period-to-date reporting, 30-9
  - sorting on, 36-44
- @LEVMBRS function, 26-37
- .LIB files, 47-5
- libraries, 45-13
- .LIC files, 47-4
- License Info page (Server Information), 46-4
- licensing, 15-22, 39-1, 45-15
  - getting information for, 46-3, 46-4
  - partitioning product suite, 6-2
- line formatting, 37-17
- linear regression, with @TREND function, 32-21
- lines, skipping in rules files, 20-11, 21-14
- LINK command
  - usage, 36-33
  - usage example, 37-57
- Link Properties dialog box, 16-31
- LINK.REP, 37-58
- linked databases. *See* linked partitions
- linked files
  - displaying contents, 12-8
  - editing, 12-8
  - LRO object type, 12-2
  - restrictions, 12-3
  - specifying size, 12-4, 12-5
  - storing, 12-3
  - valid objects as LROs, 12-1
- linked object catalogs, 12-3, 15-13

## Linked Objects Browser

- changing linked partitions, 12-9
- editing objects, 12-2, 12-7

linked objects. *See* linked reporting objects (LRO)

## linked partitions

- attaching to cells, 12-2
- changing, 12-9
- creating, 6-25, 6-26, 16-31, 16-32, 16-33, 16-35
- defined, 6-10
- described, 6-23
- disadvantages, 6-26
- guidelines for selecting, 6-26, 6-27
- implementing security measures, 6-32
- mapping members, 16-35, 16-37
- port usage, 6-26
- testing, 16-41

## linked reporting objects (LRO)

- assigning access levels, 12-4
- changing member combinations, 12-9
- checking structural integrity, 42-13
- creating, 12-2
- defined, 12-1
- estimating disk space for, 15-13
- format restrictions, 12-3
- limiting size, 12-5
- overview, 12-3, 12-4
- removing from cells, 12-6
- restructuring and, 40-20
- retrieving, 12-3
- storage management, 40-5
- types supported, 12-2
- viewing, 12-6, 12-8

Linked Reporting Objects dialog box, 12-6

## links

- currency conversions, 43-24
- linked reporting objects, 12-1
- missing, 12-3
- partitioned databases, 6-23
- related databases, 6-33, 6-38
- supported types, 12-2

@LIST function, 26-37

list of members, generating, 26-36

LISTALIASES command, 11-8

LISTFILTERS command, 18-3

LISTGROUPS command, 17-9

LISTGROUPUSERS command, 17-16

LISTLINKEDOBJECTS command, 11-13, 12-9

LISTLOCATIONS command, 7-22

LISTMEMBERS command, 8-14

## lists

- expanding/contracting in Calc Script Editor, 31-45, 31-47, 31-48
- expanding/contracting in Formula Editor, 26-21, 26-24, 26-25
- generating member, 26-4, 26-36, 31-55
- multiple, with @LIST, 26-37
- referencing, 8-23, 31-52

LISTUSERS command, 17-9, 45-10

LISTVARIABLES command, 7-19

LMARGIN command, 36-9

LOADALIAS command, 11-13

LOADAPP command, 7-5, 45-6

LOADDATA command

- loading data without rules file, 23-1
- loading free form data, 20-19
- locating files, 44-4

LOADDDB command, 7-6, 23-3, 45-8

## loading

- alias tables, 11-12
- applications
  - with related database, 45-8
- calc scripts, 31-20, 31-22, 31-23
- data, 23-8, 23-12, 23-13
  - aborted transactions and, 42-12
  - dynamic calculations and, 29-21
  - for testing purposes, 5-25
  - from external sources, 5-17, 20-3, 20-8, 23-1
  - from partitioned applications, 6-18, 6-20, 16-5
  - from rules files, 20-10
  - getting information about, 46-17
  - incrementally, 33-3
  - missing values and, 33-38
  - optimizing, 24-7, 24-9, 24-10, 24-12
  - overview, 7-3, 20-1, 20-2, 20-8, 24-7
  - prerequisites, 17-38, 23-2
  - restrictions, 20-5, 20-6, 20-15, 20-17
  - supported formats, 23-2
  - tips, 23-17, 23-18
  - troubleshooting problems with, 24-1
  - unauthorized users and, 13-46
- data files, 15-3

- loading (*Continued*)
  - data sources, 13-2, 14-27, 20-8, 24-11
    - in Data Prep Editor, 21-6, 21-7, 21-8
    - prerequisites, 23-2
    - troubleshooting problems, 24-1, 24-6
  - error log files, 24-5
  - failed records only, 24-5
  - outlines, 8-14
  - output files, 39-11
  - report scripts, 35-15, 35-16
  - rules files, 14-28, 23-9
    - prerequisites, 23-2
  - sample applications (problems with), xliv
  - specific fields only, 20-17
  - specific records, 23-17
  - spreadsheets, 23-2, 23-5, 23-18
    - multiple, 23-7
  - SQL data sources, 23-2, 23-4
    - troubleshooting problems with, 24-6
  - subsets of data, 34-1, 39-9
  - text files, 23-2, 23-5
    - multiple, 23-7
  - update log files, 48-8
- local access, 6-13
- local currency, 43-1
- locales
  - See also* currency conversions
  - dimensions defining, 5-20, 9-11
  - partitioned applications and, 6-9
- locating
  - end-of-file markers, 24-6
  - members in Calc Script Editor, 31-46
  - members in Formula Editor, 26-22
  - specific values, 4-6, 26-44
  - text
    - in report scripts, 35-17
  - text in formulas, 26-19
  - text in report scripts, 35-18
- location aliases
  - advantages, 7-19
  - editing or deleting, 7-22
- Location Aliases dialog box, 7-20
- lock files, 47-4
- Lock Manager, 40-2
  - overview, 40-6
- Lock Partition Definition Files option, 16-3
- locking caches into memory, 41-12
- locking errors, 46-40, 47-14
- locks, 47-14
  - applying, 17-37, 33-22, 33-23, 42-5, 46-40
  - committed access and, 42-7, 42-9
  - contention, 46-14
  - displaying, 17-37
  - generating reports and, 35-11
  - loading data, 20-24
  - managing, 40-6
  - maximum time to hold, 17-37
  - partition definition files, 16-3
  - removing, 17-35, 17-37, 46-40, 47-15
    - automatically, 17-37
  - setting for new users, 17-10, 17-13, 17-18
  - time-out settings, 17-31
  - types described, 42-5
  - uncommitted access and, 42-6
  - wait intervals, 42-7, 42-9
- .LOG files, 47-2
- log files
  - application events, 46-28
  - calc scripts, 31-9
  - captures failing, 23-12
  - clearing contents, 46-29, 46-32
  - controlling contents, 46-30
  - data load errors, 23-12
  - deleting, 46-29, 46-32
  - dimension builds, 14-30
  - dynamically calculated members, 29-11
  - maintaining, 7-4
  - moving through, 46-29, 46-32
  - outline changes, 16-48, 40-19
    - enabling/disabling, 46-38
    - overview, 46-35
    - restructuring and, 40-20
    - updating, 16-53
  - Server Agent activity, 45-4, 45-16
  - server events, 46-31
  - spreadsheet changes, 48-8
  - system errors, 46-26, 46-27
  - updating, 16-53
  - viewing contents, 29-10, 29-11, 31-9, 46-28, 46-31
  - viewing retrieval factor, 29-10

Log Viewer

- checking calculations, 31-9
- displaying
  - dynamically calculated members, 29-11
  - retrieval factor, 29-11
- displaying server events, 46-32
- opening, 46-29

logical conditions, 5-31, 26-29

*See also* Boolean expressions

LOGIN command, 44-7, 45-6, 45-8

login scripts (ESSCMD), 44-7

logins, 23-4

- entering user name and password, 44-7, 45-2
- limiting attempts, 17-39
- partitioned applications, 6-6
- process overview, 45-14
- setting up, 16-7

logouts

- from Application Manager, 17-33, 17-39
- from Server Agent, 45-4, 45-10

LOGOUTUSER command, 45-4, 45-10

long file names, 47-18

LOOP/ENDLOOP command, 31-12, 32-17

loops, 32-17

losing connections, 16-62

loss, tracking, 5-3

Lotus 1-2-3 spreadsheets. *See* Spreadsheet Add-in

LRO catalog, 40-5

.LRO files, 12-3, 40-5, 47-2

LRO Manager, 40-2

- overview, 40-5

LRO. *See* linked reporting objects (LRO)

@LSIBLINGS function, 26-37

.LST files, 47-2

.LWP files, 47-5

## M

M, time balance codes in data source, 14-25

Main Database (defined), 43-3

main window (Application Manager), 7-7

maintaining applications, 2-4

maintaining data integrity, 17-37

maintenance tasks, 46-18, 48-1

management privileges, 17-3

mapping

- area-specific partitions, 16-42, 16-44
  - in Partition Wizard, 16-46
- attributes, 16-28
- cells, 6-3
- columns with no members, 22-21
- data source/data target members
  - linked partitions, 16-35, 16-37
  - replicated partitions, 16-11, 16-17
  - transparent partitions, 16-25, 16-27
- databases with location aliases, 7-19
- fields to dimensions, 22-12
- files (.TXT), 16-19
- member fields to outlines, 20-15
- members, 16-11
- members to member fields, 20-5, 20-14, 20-15
- members with different names, 16-12
- partitioned members, 6-6
- partitions
  - defining manually, 16-27, 16-37
  - replicated partitions, 6-11
  - specific fields only, 22-2, 22-3, 22-4
  - transparent partitions, 6-16

Mappings page (Partition Wizard)

- defining linked mappings, 16-36
- defining replicated mappings, 16-16
- defining transparent mappings, 16-26

margin (profit), 26-2

margins, 36-9

market share (calculation example), 32-6

marketing sample reports, 37-6

marking blocks as clean and dirty, 34-2, 34-6

markup, 26-28

MASK command

- entering in report scripts, 39-13, 39-16
- overriding, 36-23
- usage, 36-27

masks, 36-23, 36-27

@MATCH function, 26-37

Match Whole Word option

- Formula Editor, 26-23

matching case, 22-15, 26-19, 26-23, 31-42, 35-17

matching members, 26-30, 31-55

mathematical calculations, 26-4

mathematical functions, 26-4, 26-38

- mathematical operations
  - currency conversions, 43-25
  - formulas and, 5-31
  - missing values and, 33-35
  - performing on fields, 22-22
  - performing on members, 9-18
  - prerequisite for, 22-23
  - report scripts, 36-16, 36-21
  - specifying in data source, 14-25
- mathematical operators, 5-31, 26-3
  - inserting in calc scripts, 31-39
  - inserting in formulas, 26-16
- matrix-style access, 4-2
- @MAX function, 26-38
- Max member, Attribute Calculations dimension, 10-33
- @MAXRANGE function, 26-43
- @MDALLOCATE function, 26-42
- @MDANCESTVAL function, 26-44
- .MDB files, 47-4
- @MDPARENTVAL function, 26-45
- @MDSHIFT function, 26-44
- .MDX files, 47-2
- Measures dimension (example)
  - calculating period-to-date values, 27-1
  - changing for currency conversions, 43-8
- media failure, 48-5
- @MEDIAN function, 26-41
- median, calculating, 26-41
- member codes
  - property tags, 14-23
  - specifying, 14-25
- member consolidation properties
  - described, 9-16
  - setting, 9-18, 14-24
- member descriptions (in Hyperion MBA). *See* alias tables
- member fields
  - defined in data source, 20-2
  - duplicate members and, 20-21
  - entering data, 20-5
  - formatting, 20-14
  - formatting rules, 20-5
  - in data source, 20-2
  - invalid, 20-6
  - mapping requirements, 20-5, 20-14, 20-15
- member lists
  - calculating subset of, 31-55
  - generating, 26-4, 26-36, 31-55
  - referencing, 8-23, 31-52
- member names
  - case sensitivity, 8-24
  - maximum length, 8-15
- Member Properties dialog box
  - creating aliases, 11-4, 11-5
  - creating UDAs, 9-26
  - defining data storage properties, 9-21
  - renaming members, 8-10
  - setting consolidation properties, 9-19
  - setting two-pass calculations, 9-16
  - setting variance reporting tags, 9-10
  - tagging for Dynamic Calculations, 9-21
  - tagging members as label only, 9-22
- member properties display option, 8-25
- member selection
  - commands, 36-27
    - sorting members and, 36-44
- Member Selection dialog box
  - defining linked areas, 16-33, 16-34
  - defining replicated areas, 16-9, 16-10
  - defining transparent areas, 16-22, 16-23
  - mapping partitioned members, 16-17, 16-27, 16-37
- member set functions, 26-4, 26-36
  - applying to subsets of members, 31-55
  - described, 26-4
  - generating lists, 26-36
  - in filter definitions, 18-10
  - inserting in formulas, 26-17
- Member Specification dialog box
  - creating dynamically calculated members, 29-22, 29-23
- members, 3-3, 3-4
  - See also* shared members
  - adding, 6-13, 13-5
    - as children of specified parent, 13-16
    - as siblings, 13-13, 13-15
  - example for, 3-13, 3-16
  - for currency conversions, 43-22, 43-25
  - guidelines for, 5-27
  - restrictions for, 3-5

members (*Continued*)

- through header information in the data source, 21-13
- to dimensions, 8-19, 9-23, 13-13, 13-15, 13-16
- to member fields, 20-14, 20-15
- to outlines, 8-15, 13-12
- adding comments about, 9-27
- adding to report scripts, 36-27, 37-57
  - in precise combinations, 36-32
  - with common attributes, 36-37
- applying skip properties, 9-6
- assigning
  - alias combinations, 11-5
  - aliases to, 11-2, 11-3
  - properties to, 5-19, 5-21
  - values to combinations, 26-46
- assigning aliases to, 11-4
- associating base dimension members with attributes
  - in Outline Editor, 10-28
- associating formulas with, 9-28
- associating with report scripts, 36-4
- Attribute Calculations dimension, 10-20
- attribute dimensions, 10-5
  - naming, 10-15
  - prefixes and suffixes, 10-16
  - preventing creation, 14-11
  - resulting from formulas, 10-13
- avoiding naming conflicts, 14-10
- calculating across multiple parents, 9-23
- calculating relationships between, 5-31
- caution for sorting with shared, 8-21
- changing, 14-10
- changing combinations for linked objects, 12-9
- changing properties, 14-23
- clearing, 22-25
- containing no data, 5-21
- creating dynamic for time series, 30-7, 36-32
- data distribution among, 4-2
- default operator, 5-26
- defined, 3-3
- defining calculation order for, 28-4, 28-5
- defining consolidations for, 28-5, 28-7
- deleting from attribute dimensions, 14-19

members (*Continued*)

- dependent on others, 9-15
- displaying
  - combinations, 4-4, 4-7
  - in outlines, 8-3, 29-6
  - in reports, 36-43
- duplicate, 20-21
- dynamically building, 13-3, 13-12, 13-13, 13-15, 13-16, 14-24
- dynamically calculating, 29-2, 29-3
  - restrictions, 29-5
- excluding from consolidation, 9-18
- getting values for specific combinations, 26-44
- grouping, 9-22, 24-8
- inserting in calc scripts, 31-43, 31-44
- inserting in formulas, 26-20
- irrelevant across dimensions, 5-15
- leaving unsorted, 14-11
- mapping names to dimensions, 22-12
- matching specified, 26-30, 31-55
- missing in data sources, 16-43, 20-13
- moving in outlines, 8-22
- moving to new parents, 14-10
- names as range of values, 20-19, 20-20, 26-33, 26-35
- naming, 8-15, 8-19, 20-5
  - maximum character length, 11-4
- nesting in reports, 36-7
- numbering in data sources, 13-5
- of attribute dimensions
  - sorting, 8-21
- ordering in dense dimensions, 4-6
- partitioning, 6-14
- relationships described, 3-5, 3-6
- removing from data sources, 14-11
- renaming, 8-10
- replicating, 6-13, 6-14
- report example for missing, 37-45
- searching in the Calc Script Editor, 31-46
- searching in the Formula Editor, 26-22
- selecting for dynamic calculations, guidelines
  - for, 29-13, 29-15
- selecting multiple, 8-22
- sharing identical values, 5-28

- members (*Continued*)
  - sorting, 8-21, 14-11
  - sorting in reports, 36-44
  - specifying as case-sensitive, 8-24
  - storing, 5-21
  - testing for, 26-30
  - unique combinations for, 4-4
  - unspecified, 18-19
  - with no data values, 9-22
  - with non-changing names, 36-39
- member-specific formatting commands, 36-9
- memory, 33-4, 38-2
  - checking available, 46-6
  - clearing, 45-8, 46-29, 46-32
  - dynamically calculated values and, 29-8
  - estimating requirements, 15-14
  - index cache size and, 24-10
  - index size and, 4-11, 4-12, 4-13
  - setting cache size, 15-16, 33-14, 33-18
    - first-time calculations, 33-21
  - shortage, 48-6
  - storing data, 40-5
  - swapping, 40-8
- memory buffers, 15-2, 15-14
- memory, locking caches into memory, 41-12
- menus
  - quick reference to commands, 46-18
  - typographic conventions, xlv
- @MERGE function, 26-37
- merging member lists, 26-37
- messages
  - displaying for calculations, 33-4, 33-23
  - example for displaying, 31-14
- metadata, 40-20
- metadata (in Hyperion MBA). *See* outlines
- MIDDLE.REP, 37-22
- migration, 2-1, 41-2
- @MIN function, 26-38
- Min member, Attribute Calculations dimension, 10-33
- minimizing resources, 4-13
- Minimum Database Access options (Application Settings), 17-32
- @MINRANGE function, 26-43
- minus signs (–)
  - as codes in data source, 14-25
  - in data fields, 20-4
  - in dimension and member names, 8-15
  - in member names, 20-5
  - in names in scripts and formulas, 8-16
  - in report scripts, 36-3
- miscellaneous text, 20-2
- mismatches, 42-13
- MISS\_LBL.REP, 37-5
- MISSING displayed in cells, 4-7
- missing links, 12-3
- missing members, 16-43, 20-13
- missing values, 4-7, 37-56
  - adding place holder for, 22-15
  - aggregating, 33-36
    - effects on calculation order, 28-15, 28-16, 28-18, 28-20
  - averages and, 30-5
  - calculating, 31-13, 31-53, 33-35
  - caution for data loads and, 23-17
  - caution for dimension fields, 20-13
  - formatting in reports, 36-14
  - handling, 9-6, 14-25, 23-17
  - identifying in data fields, 20-16
  - in calculations, 33-35
  - inserting into empty fields, 20-4
  - loading to parents, 33-38
  - optimal entry for, 24-12
  - overriding default for, 30-5
  - replacing with labels, 36-25
  - reporting samples, 37-2, 37-4
  - skipping, 9-7, 30-5
  - sorting data with, 36-49
  - testing for, 27-5
  - viewing with Personal Essbase, 39-12
- MISSINGTEXT command
  - usage, 36-25
  - usage example, 37-5
- mission-critical databases, 6-8
- @MOD function, 26-38
- @MODE function, 26-41
- mode, calculating, 26-41

Modifications page (Database Information), 46-17

Modify Outline option  
caution for disabling, 23-11

modifying  
*See also* editing  
access privileges, 17-22, 17-27, 17-28, 17-31  
alias table names, 11-9  
aliases, 14-10  
calc scripts, 31-20, 31-21, 31-22  
consolidations, 5-21  
data, 6-11, 22-22, 26-48  
prerequisite for, 17-37  
data values, 22-22  
default storage properties, 5-21  
dimension names, 8-10  
dimension properties, 13-4, 14-6, 14-10, 14-23  
dimensions, 14-10, 40-27  
default configurations, 8-9  
for currency conversions, 43-8  
formulas, 14-11, 26-13  
headings in reports, 36-8  
Hyperion Essbase kernel settings, 41-2, 41-3  
scope and precedence, 41-4  
with Application Manager, 41-5  
with ESSCMD, 41-3, 41-7  
linked partitions, 12-9  
member combinations for linked objects, 12-9  
member names, 8-10  
members, 14-10  
outlines, 8-1, 16-47, 16-52, 40-15  
caution for, 8-21, 8-22, 23-10  
dynamically, 13-3  
for currency conversions, 43-7  
with rules files, 23-10  
passwords, 17-13  
report layouts, 36-26  
security settings, 17-12, 17-21  
system password, 45-5, 45-11

modules, 47-6

modulus, calculating, 26-38

monitoring  
applications, 45-16, 46-41  
calculations, 33-4

month-to-date calculations, 30-9

@MOVAVG function, 26-42

Move Field dialog box, 22-5

moving  
fields, 22-5  
members and dimensions, 8-22  
members to new parents, 14-10  
moving average, calculating, 26-42  
moving between databases, 6-23, 6-25  
moving maximum, calculating, 26-42  
moving median, calculating, 26-42  
moving minimum, calculating, 26-42  
moving through log files, 46-29, 46-32  
@MOVMAX function, 26-42  
@MOV MED function, 26-42  
@MOV MIN function, 26-42  
MS Access databases. *See* SQL databases  
M-T-D time series member, 30-9, 30-14  
multidimensional arrays, 4-6  
multidimensional models, 3-2  
conceptual overview, 3-1  
data distribution in, 4-2  
storage requirements, 3-8  
multi-line column headings, 36-17  
multiple  
layout report, 37-36  
partitions, 6-4  
transactions, 42-4  
multiple-pass calculations, 34-3, 34-4, 34-8  
examples, 34-14, 34-15, 34-16  
usage overview, 34-13  
multiplication  
setting data consolidation properties, 9-18  
multiplication operators, 9-17  
multithreading, 1-3, 1-4, 45-15  
setting number of threads, 45-15  
multi-user environments, 1-3  
performance considerations, 33-23  
running ESSCMD in, 44-5

## N

N, never allow data sharing code in data source,  
14-25

Named Pipes connections, 45-14

names  
*See also* column headings; field names; aliases  
NAMECOL command, 36-13  
NAMESON command, 36-14

- NAMEWIDTH command, 36-13
- naming
  - batch files, 44-9
  - calc script files, 31-26, 31-28
  - dimensions, 8-15, 8-18, 14-3
  - ESSCMD script files, 44-9
  - fields, 22-12
  - filters, 18-5
  - generations, 3-8, 8-23, 14-8
  - levels, 3-8, 8-23, 14-8
  - members, 8-15, 8-19, 11-4, 20-5
  - members of attribute dimensions, 10-15
  - rules files, 21-25
  - shared members, 9-23
- naming conflicts, 14-10, 36-2
- naming conventions
  - aliases, 11-2, 11-4
  - applications, 7-10, 7-12
  - case sensitivity, 8-15, 8-24
  - databases, 7-12
  - dimensions, 8-15
  - fields, 22-13
  - generations and levels, 8-24
  - members, 8-15, 20-5
  - passwords, 17-13
  - report files, 35-25
  - rules files, 21-25
  - UNIX files, 47-16
- navigating between databases, 6-23, 6-25
- negative numbers, formatting in reports, 36-23, 36-26
- negative values
  - flipping, 22-27
  - in data fields, 20-4
  - variance as, 26-39
- nesting
  - column headings, 36-4
  - columns in reports, 37-6
  - dimensions, 38-8
  - formatting commands, 36-2
  - formulas, 31-51
  - IF statements, 26-7, 31-16
  - members in reports, 36-7
  - quotation marks in ESSCMD commands, 44-2
- net present values, 26-46
- network administrators. *See* administrators
- networks, 1-4, 45-13
  - optimizing resources, 6-8, 6-12
  - reducing traffic during data load, 24-11
  - securing, 17-1
  - transferring data via, 6-9, 6-18
- never share property, 5-22
  - description, 9-20
  - setting in data source, 14-25
- Never Share tag, 9-24
- New button (Application Manager), 7-8
- New Group dialog box, 17-15
- new line
  - as command separator, 36-2
  - as file delimiter, 20-6
- New User dialog box, 17-5, 17-10
- new users, 2-1
- NEWPAGE command, 36-10
- @NEXT function, 26-44
- No Conversion property, 9-10
  - currency conversions and, 43-10
- no loads, 23-16
- #NOACCESS value, 17-29
- node, 8-3
  - See also* branches; trees
- NOINDENTGEN command
  - usage, 36-24
  - usage example, 37-21
- non expense property, 5-31, 9-9
- None access level, 17-23, 17-25, 17-30, 18-2
- None Currency Conversion property, 9-10
- none time balance property, 9-5
- non-typical data sets, 4-13
- NOROWREPEAT command, 36-15
- NOSKIPONDIMENSION command, 36-26
- NOT operator, 36-32, 36-33
- Note dialog box, 16-6, 16-21, 16-32
- notes
  - adding to databases, 7-13
  - adding to partitions, 16-6, 16-21, 16-32
  - annotating to data cells, 12-2
  - display options, 8-25
  - storing, 12-3
  - viewing in Application Manager, 12-8
- no-wait I/O, 40-2
- .NP files, 47-5
- @NPV function, 26-46

- level references build method
      - null processing, 13-10
    - numbering
      - columns in reports, 36-18
      - members in data sources, 13-5
      - report pages, 36-24
    - numbers
      - calc scripts and, 31-15
      - calculated columns, 36-21
      - formatting, 36-23, 36-27
      - formulas and, 26-6
      - in names, 20-5
      - in source data fields, 20-4
      - options for loading, 21-16, 21-17
      - rounding, 24-12, 26-38
      - truncating, 26-39
    - numeric
      - attribute dimensions
        - sorting members in outline, 8-21
        - tagging, 9-14
      - attributes
        - defined, 10-6
        - defining member names in ranges, 10-23
        - duplicate values, 10-16
        - size of ranges, 14-18
      - fields
        - with commas, 20-4
      - fields in data source, 20-4
      - parameters, 44-3
        - entering in ESSCMD scripts, 44-8
      - numeric parameters, 44-2
      - Numeric Range Rules dialog box, 14-18
      - numerical ordering (generations), 3-7
      - NUMERICPRECISION setting, 38-5
- O**
- O, label only code in data source, 14-25
  - object handles, 12-3
  - object references, 12-3
  - objects, 47-6
    - See also* linked reporting objects
    - copying, 47-12
    - creating filters for, 18-5
  - objects (*Continued*)
    - deleting, 47-13
    - linking to cells, 12-1
    - locking, 46-40, 47-14
    - renaming, 47-13
    - unlocking, 46-40
    - .OCL files, 47-2
      - clearing, 40-19
    - .OCN files, 47-2
    - .OCO files, 47-2
    - ODBC drivers, 47-4
    - ODBC files, 47-4
    - OFFCOLCALCS command, 36-16
    - OFFROWCALCS command, 36-20
    - OFSAMEGEN command
      - usage, 36-28
    - OLAP, 3-1
      - defined, 3-1
      - fundamentals, 2-2
      - getting started tips, 2-2
    - OLAP Server, 1-1
      - installing, xliii
    - OLAP-aware modules, 47-6
    - .OLB files, 46-39, 47-2
    - .OLG files, 46-35, 47-2
    - OLTP, 3-1
    - ONCOLCALCS command, 36-16
    - Online Analytical Processing. *See* OLAP
    - Online Transaction Processing. *See* OLTP
    - ONROWCALCS command, 36-20
    - ONSAMELEVELAS command
      - usage, 36-28
    - Open button (Application Manager), 7-8
    - Open Client Data Files dialog box
      - importing outlines, 8-14
      - opening data sources, 21-6
      - opening spreadsheets, 21-7, 23-7
      - opening text files, 23-7
      - updating dimensions, 14-28
    - Open Client Object dialog box
      - selecting calc script files, 31-23
    - Open Server Data File Object dialog box
      - opening data sources, 21-5
      - opening spreadsheets, 21-7, 23-6
      - opening text files, 23-6
      - updating dimensions, 14-27

- Open Server Object dialog box
  - opening report scripts, 35-14
  - selecting calc script files, 31-21
- open tasks, 45-16
- opening
  - applications, 7-7, 7-8
  - Calc Script Editor, 31-18
  - calc script files, 31-32, 31-34
  - Data Prep Editor, 14-3, 21-3
  - data sources, 21-5
  - databases, 7-7, 7-8
  - Formula Editor, 9-29, 26-12
  - Log Viewer, 46-29
  - outlines, 8-2
    - caution for, 8-8
  - Partition Manager, 16-2
  - Report Editor, 35-2
  - report script files, 35-14, 35-16
  - rules files, 14-28, 21-3
  - sample applications (problems with), xliv
  - spreadsheets, 21-7, 23-5
  - SQL data sources, 20-3, 21-8
  - text files, 21-5
- operating system
  - information, 46-3, 46-6
  - recovery, 48-6
- operations, 47-5
  - See also* transactions
  - aborted transactions and, 42-12
  - automating routine, 2-4
  - canceling
    - archiving, 48-3
    - calculations, 25-11
  - canceling ESSCMD, 44-3, 44-8
  - causing corruption, 48-5
  - displaying field, 20-10
  - displaying replace, 20-11
  - failed, 42-11
  - maintaining data integrity for, 17-37
  - missing values and mathematical, 33-35
  - not supported, 3-2
  - privileges and routine, 17-3
  - restructure types defined, 40-15
  - undoing, 22-11, 31-36
  - viewing information about, 46-3, 46-17
- operators
  - See also specific operator*
  - calc scripts and, 31-15, 31-40
  - consolidation listed, 9-17
  - creating Boolean expressions with, 36-33
  - cross-dimensional, 4-7, 33-11
    - inserting in calc scripts, 31-40
    - inserting in formulas, 26-3, 26-16
    - overview, 26-46, 33-10
    - usage examples, 3-10, 26-47
  - default for members, 5-26
  - display options, 8-25
  - formulas and, 5-31, 26-3, 26-6, 26-17
  - inserting in formulas, 26-15
  - mathematical, 5-31, 26-3
    - calc scripts, 31-39
    - formulas, 26-16
  - order of precedence, 9-17
  - resetting in report scripts, 36-21
  - unary, 5-26, 5-27
    - usage overview, 28-5, 28-7
- optimizing
  - access, 6-2, 6-8
  - cache, 33-3, 33-22
  - calculations, 5-24, 33-1, 33-21
    - with bottom-up calcs, 33-12
    - with calc scripts, 31-50
    - with calculator cache, 33-15
    - with dynamic calculations, 29-5, 29-13
    - with Intelligent Calculation, 34-1
    - with interdependent values, 26-33
    - with two-pass calculations, 33-24
  - data analysis, 12-1
  - data loading, 24-7, 24-9, 24-10, 24-12, 33-3
  - data manipulation, 13-1
  - data sources, 24-11, 24-12
  - disk read/writes, 24-10, 24-11
  - indexes, 33-32, 33-33
  - network resources, 6-8, 6-12
  - outlines, 24-10, 33-4
  - performance, 5-4, 6-2, 29-6, 33-1, 33-2, A-1
  - queries, 5-23, 10-14
  - readability, 11-1
  - replication, 6-13
  - reports, 12-1, 36-29, 38-1

optimizing (*Continued*)

- restructures, 40-18, 40-19
- retrieval, 29-10, 29-12
- sparse dimensions, 24-8, 24-10
- storage, 6-8
- transparent partitions, 6-20

## optional parameters, 44-2

## options

- See also* display options
- application access, 17-23
- calculations, 25-7, 31-13
- dimension builds, 23-11
- dimension updates, 14-10
- dynamic builds, 13-4, 14-20
- generating currency outlines, 43-19
- global access, 17-32, 17-33, 17-35
- Hyperion Essbase kernel, 41-2, 41-3, 41-4
- isolation levels, 41-16
- level and generation numbering, 14-17
- password management, 17-40
- restructuring outlines, 8-8
- saving partition definitions, 16-41
- setting up rules files, 13-3
- viewing
  - currently set installation, 46-4
  - linked reporting objects, 12-7

## OR operator, 36-32

- select/reject criteria, 21-16, 21-18

Oracle databases. *See* SQL databases

## ORDER command

- usage example, 37-21, 37-33

## ORDERBY command

- entering in report scripts, 36-46, 36-47, 36-48
- precedence, 36-46
- usage, 36-45
- usage example, 37-55

## ORDERBY.REP, 37-56

## ordering

- cells in blocks, 4-6
- data blocks, 28-3
- data values, 36-45, 36-46
  - example, 37-55
- dimensions in outlines, 28-6
- fields, 22-5, 22-11
- members in dense dimensions, 4-6

ordering (*Continued*)

- members in outlines, 4-6, 14-10
- output values, 36-47
- ordinary user privileges, 17-6
- organizational databases, 5-16
- organizing data (tutorial), 3-13
- .OTL files, 40-16, 40-17, 47-2
- .OTM files, 47-2
- .OTN files, 40-16, 47-3
- .OTO files, 47-3
- OUTALT command, 36-42
- OUTALTMBR command
  - entering in report scripts, 36-43
  - usage, 36-42
- OUTALTNames command
  - entering in report scripts, 36-43
  - usage, 36-42
  - usage example, 37-21
- OUTALTSELECT command
  - usage, 36-42
- outline
  - attribute prefixes and suffixes, 9-13
- Outline button (Application Manager), 7-8
- outline calculation, 25-2
- outline change log files, 40-19
  - contents described, 16-48, 46-37
  - enabling/disabling, 46-38
  - format example, 46-36
  - overview, 46-35
  - restructuring and, 40-20
  - setting size, 46-39
  - updating, 16-53
- outline design
  - attribute dimensions, 5-15
  - performance considerations, 5-23, 10-14
- Outline Editor, 8-1, 40-14
  - adding comments, 9-27
  - adding dimensions, 8-18
  - applying time balance tags, 9-8
  - associating attribute dimensions with base
    - dimensions, 10-26
  - associating base dimension members with
    - attributes, 10-28
  - Attribute Calculations dimension, 10-12

- Outline Editor (*Continued*)
  - changes and restructuring impact, 40-27
  - components, 8-4
    - enabling/disabling, 8-25
  - creating
    - aliases, 11-4
    - aliases for member combinations, 11-5
    - dynamically calculated members, 29-22
    - outlines, 4-8
    - shared members, 9-25
    - UDAs, 9-26
  - creating dynamically calculated members, 29-22
  - customizing, 8-24, 8-25, 8-26, 8-27
  - defining
    - accounts dimension type, 9-4
    - attribute dimension type, 9-13
    - attributes, 10-25
    - country dimension type, 9-11
    - data storage properties, 9-20
    - dimension type, 9-10
    - time dimension type, 9-3
  - defining dimension type
    - Currency Partition, 43-15
  - importing outlines to, 8-13
  - removing dynamically calculated members, 29-23
  - renaming dimensions and members, 8-10
  - selecting multiple members, 8-22
  - setting consolidation properties, 9-18
  - setting two-pass calculations, 9-16
  - setting variance reporting tags, 9-9
  - starting, 8-3
  - tagging members as label only, 9-22
  - updating dimensions, 14-26
  - viewing outline members, 8-3
- outline files, 40-16
  - copying, 39-2, 39-4, 39-6
  - creating, 39-7
  - cross-platform compatibility, 47-16
  - saving, 39-6
- outline synchronization
  - described, 16-47
  - shared members, 16-53
- Outline Synchronization Editor, 16-51
- Outline Update dialog box, 14-27
- OUTLINECHANGELOG parameter, 40-20, 46-35, 46-38
- OUTLINECHANGELOGFILESIZE parameter, 46-39
- outline-only restructure, 40-15
- outlines, 7-3, 38-8
  - accessing, 38-8
  - adding alias tables, 11-7, 11-9
  - adding dimensions, 3-18, 4-8, 4-11, 8-15, 8-18
    - performance considerations, 33-3, 33-8
    - restructuring and, 40-28
  - adding members, 8-15, 13-12
  - applying changes, 16-50, 16-52
  - associating
    - calc scripts with, 31-44
    - member fields with, 20-15
    - rules files with, 14-6, 14-9, 14-14, 21-22
  - associating filters with, 18-4
  - bottom-up ordering, 9-15
  - changes impacting restructuring, 40-21, 40-27
  - changing, 8-1, 16-47, 16-52, 40-15
    - caution for, 8-21, 8-22, 23-10
    - dynamically, 13-3
    - with rules files, 23-10
  - components of, 3-4
  - controlling location of, 6-8
  - copying, 8-4, 47-12
    - with Application Manager, 39-5
  - creating, 3-3, 3-4, 4-8, 5-19
    - example for, 3-13
    - for currency conversions, 43-17, 43-18, 43-21
    - from data sources, 13-2
    - guidelines for, 5-14
    - prerequisites for, 5-4
    - with Outline Editor, 8-1, 8-2
  - creating formulas for, 26-8
  - customizing view, 8-25
  - defined, 3-4, 5-2
  - drafting for single-server databases, 5-17
  - expanding/contracting, 8-3
  - importing/exporting, 8-10, 8-11, 8-13
  - improving readability, 11-1
  - loading, 8-14
  - member relationships described, 3-5, 3-6
  - modifying for currency conversions, 43-7

outlines (*Continued*)

- naming
  - dimensions and members, 8-15
  - generations and levels, 8-23
- opening existing, 8-2
  - caution for, 8-8
- optimizing, 24-10, 33-4
- optimum order of dimensions, 10-14
- ordering dimensions, 28-6
- ordering members, 4-6, 14-10
- rearranging members and dimensions, 8-20
- removing, 47-13
- removing items, 3-19, 11-10
- renaming, 47-13
- repeating elements, 5-14
- restructuring, 8-8
  - prerequisites for, 17-38
- rules, 8-5
- saving, 40-27, 40-28, 40-29
  - caution for, 8-8
  - description, 8-5
  - instructions, 8-7
  - on server or client, 8-12
- sharing members, 13-32, 13-42, 13-43
  - caution for placing, 9-23
- synchronizing, 6-2, 6-25
  - process summarized, 16-47, 16-48
  - warning for not applying changes, 16-49
  - with Application Manager, 16-50
  - with Partition Wizard, 16-4
  - with report scripts, 38-8
- top-down ordering, 9-17
- tracking changes, 16-48
- updating, 14-26, 23-12
- verifying, 8-5, 8-7
- viewing
  - changes to, 46-35
  - dimensions in, 8-3
  - dynamically calculated members, 29-6
  - update information, 46-17
  - with unbalanced hierarchies, 36-37

OUTMBRALT command, 36-42

OUTMBRNames command, 36-42

## output

- displaying in columns, 37-22
- formatting, 36-15
- options, 35-21
- ordering, 36-47
- reporting example, 37-37
- security information, 45-12
- selecting specific values for, 36-46
- sorting, 36-46

OUTPUT command, 36-14

output files, 39-1

- See also* log files
- loading, 39-11
- saving, 39-9

overhead, 15-8, 15-11, 40-8

- checking compression ratio, 40-11
- partitioned applications, 15-12

overlapping partitions, 6-5

overriding

- column groupings in reports, 36-8
- data filters, 16-5, 25-12
- default calculation order, 28-4
- default calculations, 31-2
- file locks, 46-40
- incremental restructuring, 40-19
- security and access levels, 17-33

overwriting error log files, 46-27

overwriting existing values, 30-6

- for currency conversions, 43-26
- with values in data source, 22-23, 23-17

ownership, 6-9

**P**

.PAG files, 40-16, 47-3

page breaks

- in report scripts, 36-10
- suppressing, 36-14, 36-23

PAGE command

- entering in report scripts, 36-4
- in page layout, 36-4
- usage example, 37-36

page files, 40-12

- See also* data files

- page headings
  - adding to complex reports, 36-4
  - adding to free-form reports, 35-29
  - customizing, 36-6, 37-27
  - defined, 35-7
  - display options, 36-6
  - forcing immediate display, 36-6
  - suppressing, 36-14
  - using attributes in, 36-5
- page layout commands, 36-3, 36-6
- page layouts
  - See also* reports
  - adding titles, 36-24
  - centering data, 36-9, 36-17
    - example, 37-27
  - changing, 36-26
  - customizing, 36-6, 36-18
    - examples, 37-23, 37-27
  - formatting, 36-8, 36-9, 36-10, 36-16, 36-22
    - problems with, 36-51
  - formatting samples, 37-2, 37-17, 37-35, 37-37
  - inserting page breaks, 36-10, 36-23
  - numbering pages, 36-24
  - switching row/column display, 37-8
- PAGEHEADING command, 36-6
- PAGELength command
  - overriding, 36-23
  - usage, 36-10
- PAGEONDIMENSION command
  - usage, 36-10
  - usage example, 37-5
- .PAN files, 40-16, 47-3
- parameters
  - enclosure in quotation marks (ESSCMD), 44-2
  - entering in ESSCMD scripts, 44-8
  - referencing files, 44-3
- PARCHIL.RUL, 13-12
- PARCHIL.TXT, 13-11
- @PARENT function, 26-37
- PARENT command
  - usage, 36-28
- parent field type
  - in header records, 14-24
  - in rules files, 14-15
  - sharing members, 13-36, 13-39, 13-42
- PARENT. *See* parent field type
- parent/child references build method
  - creating multiple roll-ups, 13-44
  - creating shared members, 13-36, 13-39, 13-42
  - creating shared roll-ups from multiple data
    - sources, 13-44
  - described, 13-11
  - example rules file, 13-12
  - guidelines for using, 13-4
  - sharing members, 13-36, 13-39, 13-42
  - valid field types, 14-15
- parent/child relationships, 3-6
  - data sources, 13-3
  - defining, 13-11
  - dynamic calculations and, 29-4, 29-7, 29-9
- parentheses
  - in calc scripts, 26-29
  - in dimension and member names, 8-15
  - in formulas, 31-51
  - in names in scripts and formulas, 8-17, 36-3
  - in report scripts, 36-33, 37-58
  - indicating negative numeric values in fields, 20-4
- parents
  - as shared members, 13-33
  - assigning children to, 13-16, 13-38, 13-41
  - associating members with, 14-10
  - calculation order for outlines, 9-15
  - calculations with multiple, 9-23
  - checking for, 26-30
  - defined, 3-6
  - getting, 26-37, 26-44
  - in time dimensions, 30-1
  - loading data into, 23-17
  - loading data to, 33-38
  - rolling up, 13-32, 13-33
  - setting values as average, 9-6
  - sharing members, 13-32
  - specifying for existing members, 14-11
  - with only one child, 9-24
- @PARENTVAL function, 26-44, 32-7
- partial loads, 23-14
  - invalid members, 20-6
  - value out of range, 20-20
- partition areas
  - changing shared, 16-48
  - defined, 6-6
  - defining, 16-8, 16-22, 16-33

- partition areas (*Continued*)
  - Dynamic Time Series members in, 30-16
  - mapping to specific, 16-42, 16-44
    - in Partition Wizard, 16-46
- partition definition files, 16-1
  - creating, 16-40
  - locking, 16-3
- Partition Information dialog box, 16-58
- Partition Manager, 16-3
  - changing links, 12-9
  - creating partitions, 16-3, 16-4, 16-20, 16-31
  - defining partitioned areas
    - replicated partitions, 16-9, 16-10
    - transparent partitions, 16-23
  - editing partitions, 16-57
  - opening, 16-2
- partition size, 41-22
- Partition Wizard
  - closing, 16-41
  - defining area-specific mappings, 16-46
  - defining partitioned areas, 16-8, 16-22, 16-33
  - described, 16-3
  - editing partitions, 16-58
  - mapping linked members, 16-36, 16-37
  - mapping partitioned members, 16-17
  - mapping replicated members, 16-16, 16-17, 16-19
  - mapping transparent members, 16-26, 16-27
  - saving partition definition, 16-40
  - validating partitions, 16-38
- partitioned applications
  - accessing data, 6-10
  - adding members, 6-13
  - calculating, 26-49
    - transparent, 6-21
  - connecting to, 16-6, 16-21, 16-32
  - creating, process summarized, 6-7
  - described, 6-2
  - designing, 6-1, 6-8, 6-9
    - scenarios for, 6-33, 6-36, 6-38
  - disadvantages of, 6-9
  - estimating disk space for, 15-12
  - examples of, xliv
  - language requirements, 6-9
  - loading data, 6-18, 6-20, 16-5
- partitioned applications (*Continued*)
  - performing calculations on, 26-49
    - replicated partitions and, 6-13
    - transparent partitions and, 6-18, 6-19, 6-21, 6-22
  - retrieving data, 6-3
  - running calc scripts, 31-57
  - single-server vs., 5-1
  - troubleshooting access to, 16-63
  - types of, 6-10
  - updating, 6-7, 6-11, 6-13, 16-60
    - guidelines, 16-58
    - remote data and, 6-15
  - viewing current state, 6-7
- partitioned databases
  - accessing, 6-8, 6-9, 6-26
  - adding partitions for currency conversions, 43-4, 43-26
  - calculating, 26-49
    - transparent, 6-21
  - creating accounts for, 6-26, 6-31, 6-32
  - described, 6-3
  - dynamically calculating values, 29-25
  - filtering, 6-30, 6-31
  - implementing security measures, 6-30, 6-31, 6-32
  - linking data values, 6-23
  - maintaining, 16-1
  - restructuring, 6-19, 40-21
  - sample applications showing, 6-10
  - sharing data, 16-8, 16-22, 16-33
  - size and storage considerations, 15-12
  - storing data, 6-8, 6-9
    - sparse member combinations and, 28-3
    - with replicated partitions, 6-13
    - with transparent partitions, 6-18
  - synchronizing data across partitions, 40-21
  - synchronizing outlines, 6-2, 6-25
    - with Partition Wizard, 16-4
  - testing, 6-7, 16-41
  - time series reporting, 30-16
  - troubleshooting connections, 16-63
  - workflow, 6-7
- partitioning
  - mapping attributes, 16-28
  - using attributes in, 6-2, 16-38

- partitioning product suite, 1-2
  - licensing, 6-2
- partitions
  - See also* partitioned applications; partitioned databases, areas
  - advantages, 6-2
  - annotating, 16-6, 16-21, 16-32
  - calculating, 26-49
    - transparent, 6-21
  - controlling updates to, 6-11
  - creating, 16-1, 16-3, 16-4, 16-20, 16-31
    - for currency conversions, 43-4, 43-26
    - process summarized, 6-7
  - defined, 6-6
  - defining, 6-3, 6-8, 6-9
    - linked, 6-25, 6-26
    - multiple, 6-4
    - replicated, 6-11, 6-12, 6-13
    - transparent, 6-16, 6-18
    - with Partition Wizard, 16-3, 16-4, 16-20, 16-31
  - deleting, 16-58
  - discarding changes to, 16-41
  - dynamic calculations and, 6-13, 6-22
  - editing, 16-57
  - getting type, 6-6
  - mapping guidelines, 6-11, 6-16
  - mapping members, 16-11
  - overlapping, 6-5
  - parts described, 6-6
  - performance
    - improving, 6-20
  - port usage, 6-15, 6-23, 6-26
  - primary/secondary sites defined, 6-3
  - restructuring performance and, 40-21
  - saving definitions, 16-40
  - selecting type, 6-10, 6-15, 6-20, 6-26, 6-27
    - in Partition Wizard, 16-3, 16-5, 16-21, 16-31
  - top-down vs. bottom-up, 6-33
  - troubleshooting, 16-62
  - usage examples, 6-10
    - replicated partitions, 16-4
    - transparent partitions, 16-20
  - using attributes in, 6-28
  - validating, 16-38
  - viewing currently defined, 16-57, 16-58
- PASSWORD command, 45-5
  - usage overview, 45-11
- Password Management options (Server Settings), 17-40
- passwords
  - assigning generic, 17-10
  - changing, 17-13
  - connections, 45-11
  - entering in logins, 44-7, 45-2
  - forcing users to change, 17-10, 17-13, 17-17
  - in partitions, 6-6
  - masking, 16-7, 17-10
  - setting, 17-39
    - new users, 17-10, 17-17
    - partitioned databases, 16-6
  - system, 45-5, 45-11
- pasting
  - calc scripts into Calc Script Editor, 31-10
  - formulas into Formula Editor, 26-11
  - text into calc scripts, 31-42
  - text into formulas, 26-19
  - text into report scripts, 35-20
- path information, 46-5
- pattern matching, 36-38, 36-39
- payroll, 26-31
- percent signs (%)
  - as codes in data source, 14-25
  - in names in scripts and formulas, 8-17
  - in report scripts, 37-23
- percentages
  - allocating, 26-47
  - calculating, 28-7, 32-6
  - displaying in reports, 36-26, 37-32
  - returning variance, 26-39, 32-2
  - setting consolidation properties, 9-18
  - specifying in data source, 14-25
- performance, 1-3, 4-1
  - calculation, 33-1, 33-15, 33-18
  - checking, 15-18, 46-1
  - getting information about, 46-6, 46-7
  - linked partitions and, 6-24, 6-25
  - log files and, 46-30
  - multi-user considerations, 33-23
  - optimizing, 5-4, 6-2, 10-14, 29-6, 33-1, 33-2, A-1
  - recommended settings, A-1
  - replicated partitions and, 6-13

- performance (*Continued*)
  - restructure operations and, 40-18, 40-20
  - transparent partitions and, 6-20
- performance-related storage settings, 41-2, 41-3
  - scope, 41-4
- periodic year to date (in Hyperion MBA). *See*
  - dynamic time series members
- periods (.)
  - in application and database names, 7-12
  - in dimension and member names, 8-15
  - in names in scripts and formulas, 8-17
- period-to-date calculations, 30-8
- period-to-date values, 27-1
  - calculating, 26-46, 30-7
  - retrieving, 30-14
- Personal Essbase, 39-1
  - copying data to, 39-2
    - prerequisites, 39-7
  - copying outlines to, 39-4, 39-5, 39-6
  - creating applications and databases, 39-3
  - installing, 39-1
  - loading data, 39-9
  - loading output files, 39-11
  - viewing data, 39-12
- PIDs, finding for Hyperion Essbase applications, 45-7
- pivoting, 3-2
- platforms, 1-5
  - complete listing, 45-13
  - porting applications across, 47-15, 47-18
    - creating backups for, 48-3
    - redefining information for, 47-19
  - porting applications to UNIX servers, 47-16
- plus signs (+)
  - as codes in data source, 14-25
  - in application and database names, 7-12
  - in dimension and member names, 8-15
  - in member names, 20-5
  - in names in scripts and formulas, 8-17, 36-3
- pointers
  - data blocks, 4-5
  - member combinations, 26-46
  - shared data values, 5-28, 9-23
- populating replicated partitions, 16-60
- porting applications, 47-15, 47-18
  - creating backups for, 48-3
  - redefining server information, 47-19
  - to UNIX platforms, 47-16
- ports
  - and linked partitions, 6-26
  - and replicated partitions, 6-15
  - and transparent partitions, 6-23
  - displaying available, 45-5, 45-10
  - displaying installed, 45-5, 45-10
  - displaying list of available, 45-4, 45-10
  - freeing, 45-4, 45-10
  - licensed and multithreading, 45-15
  - protocol-specific assignments, 45-14
  - reserved, 45-15
  - running out of, 16-62
- PORTS command, 45-5
  - usage overview, 45-10
- positive values, 22-27
  - variance as, 26-39
- @POWER function, 26-38
- power down (caution), 17-33
- power failures. *See* failures; recovery
- power loss, 48-6
- precedence
  - calculations, 9-17
- precision, 38-5
- predefined Dynamic Time Series members, 30-7
  - enabling/disabling, 30-9, 30-11
  - generation names for, 30-14
  - in shared areas, 30-16
  - listed, 30-8
  - specifying aliases for, 30-13
- predefined routines, 5-31, 26-4
- prefixes
  - adding to fields, 22-20
  - assignment sequence in data build, 20-11
  - attribute member names, 9-13, 10-16, 10-17
  - member and alias names, 8-17
- Pre-image Access option, 41-16, 42-3
  - locks and, 42-7
- preventing calculation delays, 33-24
- preventing system failures, 6-8, 6-12
- primary roll-ups, 13-38, 13-40, 14-17
- Print Calc Script dialog box, 31-35

- printed documentation, xlii
  - additional resources, xlv
  - suggested audience, xli
  - typographical conventions, xlv
- printing
  - calc scripts, 31-35
  - formulas, 26-14
  - reports, 35-22, 44-15
- PRINTPARTITIONDEFFILE command, 16-58
- PRINTROW command, 36-20
- @PRIOR function, 26-44, 27-4
- privileges, 17-36
  - Application Designer, 17-35
  - applying to groups, 17-14
  - assigning
    - global, 17-30, 17-31, 17-34, 17-35
    - to users and groups, 17-5
  - assignment examples, 19-1, 19-2, 19-3, 19-4, 19-5
  - changing, 17-22, 17-27, 17-28, 17-31
  - global security settings and, 17-29
  - global types, 17-25
  - inheritance from group, 17-4
  - inheriting, 17-29, 17-30
  - insufficient, 17-29
  - layers defined, 17-1
  - levels defined, 17-2
  - linked reporting objects, 12-4
  - overriding, 17-33
  - planning for user access, 5-17
  - replicating, 17-16
  - required, 17-5
  - routine operations and, 17-3
  - setting filters and, 18-1, 18-2
  - transactions and, 42-3
  - types listed, 17-30, 18-2
  - user types, 17-5
- procedural commands, 31-14
- process IDs, finding for Hyperion Essbase
  - applications, 45-7
- processes, 45-12, 45-16
  - monitoring, 46-41
- processing requests, 1-4, 45-13
- processors, 1-3
  - calculations across multiple, 6-8
  - supported as clients, 1-4
- product and market shares, 32-6
- Product dimension (example), 4-14
- product version, 45-5, 45-12
- profit and loss, 32-17, 43-8
  - example for tracking, 5-3
- profit margins, 26-2
  - dynamically calculating, 29-17
- program files, 47-1
- programming interface. *See* API (Application Programming Interface)
- programming languages (supported), 1-5
- programming objects, 47-6
- programming-specific files, 47-5
- propagating outline changes. *See* synchronizing
- properties, 14-23
  - changing dynamically, 13-4, 14-6, 14-10, 14-23
  - consolidation, 9-16, 9-18, 14-24
  - currency conversion, 9-10
  - data storage, 5-21, 9-20
  - defining
    - caution for two-pass tags, 9-15
    - for dimensions, 9-1, 9-2
    - for members, 9-1, 9-16
  - defining for dimensions, 5-19, 14-6
  - defining for members, 9-18, 14-24
    - as label only, 9-22, 14-25
  - design checklist for, 5-22
  - dimension building
    - field types, 14-15
  - dynamic calculations, 9-21
  - setting dynamically, 14-24
  - shared member, 9-23, 9-25
  - time balance, 5-30
  - Time Series, 30-5
  - two-pass calculations, 9-15
  - variance reporting, 5-31, 9-9
- properties bar (Outline Editor), 8-25
- properties in outlines, 5-19
- property field type
  - in header records, 14-24
  - in rules files, 14-15
  - nulls and, 13-7, 13-10
  - specifying in data source, 14-25
- PROPERTY. *See* property field type
- protecting data, 42-1, 48-1
  - See also* security

@PTD function, 26-46, 27-1, 30-7  
 P-T-D time series member, 30-8, 30-14  
 PURGELINKEDOBJECTS command, 11-13, 12-9  
 PURGEOTLCHANGEFILE command, 16-56  
 purging log files, 16-53  
 PUTALLREPLCELLS command, 16-62  
 PUTUPDATEDREPLCELLS command, 16-62  
 PYRAMIDHEADERS command, 36-8, 36-17

## Q

Q-T-D time series member, 30-9, 30-14  
 quarterly values, 37-32  
 quarter-to-date calculations, 30-9  
     example, 30-8  
 queries, 3-14  
     optimizing performance, 10-14  
 question marks (?)  
     in application and database names, 7-12  
     used as wildcard, 36-38  
 QUIT command, 45-3, 45-5  
     usage overview, 45-12  
 quitting  
     *See also* closing; exiting; stopping  
     Essbase, 45-5, 45-12  
     Partition Wizard, 16-41  
 quotation marks, double ("")  
     and ESSCMD commands, 44-2  
     enclosing member names, 20-14, 22-13, 36-3  
     in application and database names, 7-12  
     in calc scripts, 31-15  
     in dimension and member names, 8-15  
     in formulas, 14-11, 26-6  
     in report scripts, 39-8  
     in scripts and formulas, 8-16, 8-17  
 quotation marks, single (')  
     in application and database names, 7-12  
     in dimension and member names, 8-15  
 QUOTEMBRNAMES command, 39-8

## R

@RANGE function, 26-37  
 range functions, 26-4, 26-43, 33-6, 33-7  
     formulas and, 31-50

range of values  
     *See also* ranges  
     calculating, 31-55  
     copying, 31-54  
     data exceeding, 20-20  
     duplicate members in, 20-21  
     getting next member, 26-44  
     iterating through, 26-44  
     member names as, 20-19, 20-20, 26-33, 26-35  
     optimizing data loads with, 24-11  
     reading multiple, 20-22  
     report member selection, 36-46, 36-48  
     setting automatically, 20-20  
 ranges  
     numeric  
         caution regarding inserting new values, 13-28  
         dimension building summary, 13-29  
     numeric attributes, 10-7, 10-23  
         automatic building, 14-18  
         building multilevel (example), 13-25  
         different sizes, 13-25  
 @RANK function, 26-41  
 ranking values, 26-41  
 ratios, 1-2  
 raw data sources, 21-3  
     viewing, 21-4  
 Read locks, 42-5  
     described, 42-5  
     with committed access, 42-2, 42-7  
     with uncommitted access, 42-7  
 Read Only privilege, 17-25  
 Read privilege, 17-30, 18-2  
 Read/Write privilege, 17-25  
 reads, 1-3  
     getting number of, 46-14  
     optimizing, 24-11  
 rearranging fields, 22-5, 22-11  
 reboot (caution), 17-33  
 rebuilding databases, 42-13  
 recalculating data, 3-19, 8-15, 8-20, 29-20  
     after exporting, 48-4  
     Dynamic Calc And Store members, 29-3  
     examples, 32-5  
     for Personal Essbase servers, 39-12

- recalculating data (*Continued*)
  - in sparse dimensions, 31-50, 34-11
  - Intelligent Calculation and, 34-4, 34-6
  - two-pass calculations and, 33-26
- reconfiguring dimensions, 14-21
- Record View Count dialog box, 21-20
- records
  - See also* rows
  - adding to data sources, 13-5, 13-8, 13-11
  - as headers in rules files, 14-23, 21-11
  - bottom-up ordering and, 13-8
  - defined, 20-1
  - defining operations for rules files, 13-2
  - defining roll-ups for, 13-38, 13-40, 14-17
  - in data source, 13-3
  - loading specific range, 23-17
  - missing from error logs, 24-2
  - rejecting for loads, 21-16
  - reloading failed, 24-5
  - selecting, 21-15, 21-16, 21-18
  - setting parent/child relationships, 13-11
  - sorting to optimize data loading, 24-8
  - top-down ordering and, 13-5
  - types in data sources, 21-11
  - viewing, 21-3, 21-20
  - with extra fields, 20-17
- recovery, 40-7, 48-2
  - calculations not completing, 17-37
  - caution for loading data and, 22-23
  - failed operations, 42-11
  - improper shutdowns, 17-33
  - managing, 40-6
  - procedures, 48-7
  - redundant data and, 42-11
  - restructuring databases, 40-17
  - server crashing, 24-5, 48-5, 48-6
    - suggested procedures, 48-7
- redefining server information, 47-19
- reducing database size, 6-13
- reducing network traffic, 6-12
- redundant data, 41-16, 42-11
- references
  - caution for, 33-10
  - data values, 3-8, 26-49, 33-6, 33-7
- references (*Continued*)
  - dynamic, 14-23, 21-13
    - record numbers in, 21-14
  - dynamically calculated members, 29-9
  - filters and updating, 18-16
  - forward calculation, 28-8
  - generation, 13-5
    - null processing, 13-7
    - sample rules file, 13-6, 13-34
    - shared members, 13-33
  - level, 13-8
    - example rules file, 13-9, 13-43
    - null processing, 13-10
    - sample rules file, 13-35
    - shared members, 13-35, 13-38, 13-40
  - linked files, 12-8
  - lists, 8-23, 31-52
  - objects, 12-3
  - parent/child, 13-11
    - example rules file, 13-12
    - sharing members, 13-36, 13-39, 13-42
  - specific values, 25-6
- refreshes
  - data, 6-13
  - dialog boxes, 46-3
- Region dimension (example), 4-14
- Reject Record dialog box, 21-17
- rejection criteria
  - defining multiple for single field, 21-17
  - defining on multiple fields, 21-18
  - example, 23-18
  - locating end-of-file markers with, 24-6
  - specifying in rules file, 20-12, 21-16
- relational databases. *See* databases
- relational storage manager, 40-20
- relationship among members, 3-6, 25-2, 28-4
  - report example for missing, 37-45
- relationship functions, 26-4
  - formulas and, 33-6, 33-7
  - Intelligent Calculation and, 34-19
- @RELATIVE function, 26-37
- reloading database files, 47-20
- reloading exported data, 48-4
- @REMAINDER function, 26-38

- remainders, 26-38
- remote locations
  - accessing, 6-8
  - manipulating data, 6-15
  - retrieving data, 6-15, 29-14, 29-25
- remote partitions. *See* transparent partitions
- remote shutdowns, 45-3
- @REMOVE function, 26-37
- Remove Unspecified option, 14-11
- REMOVECOLCALCS command
  - entering in report scripts, 36-19
  - usage, 36-16
- REMOVELOCKS command, 17-38
- REMOVEUSER command, 17-12
- removing
  - See also* clearing
  - aliases, 11-6
  - applications, 17-35, 47-8
  - calc scripts, 31-35, 31-36
  - contents from alias tables, 11-11
  - data blocks, 29-20
  - databases, 47-11
  - dimensions, 3-19
    - restructuring and, 40-29
  - filters, 18-14
  - formulas, 26-14
  - items from outlines, 3-19, 11-10
  - linked objects, 12-6
  - locks, 17-35, 17-37, 46-40, 47-15
    - automatically, 17-37
  - log files, 46-29, 46-32
  - members from data sources, 14-11
  - members from member lists, 26-37
  - objects, 47-13
  - partitions, 16-58
  - substitution variables, 7-17
  - temporary files, 40-17
  - text in calc scripts, 31-41
  - text in formulas, 26-18
  - text in report scripts, 35-20
  - user groups, 17-20
  - users, 17-11, 17-19
  - white space in fields, 22-18
- Rename Alias Table dialog box, 11-9
- Rename Application dialog box, 47-7
- RENAME command, 37-40
- Rename Database dialog box, 47-10
- Rename Filter dialog box, 18-13
- Rename Group dialog box, 17-22
- Rename Object dialog box, 47-13
- Rename User dialog box, 17-21
- RENAMEAPP command, 47-8
- RENAMEDDB command, 47-10
- RENAMEFILTER command, 18-14
- RENAMEUSER command, 17-22
- renaming
  - alias tables, 11-9
  - applications, 47-7
  - databases, 47-10
  - dimensions, 8-10
  - error log files, 24-5
  - files with FTP, 47-18
  - filters, 18-13
  - members, 8-10
  - objects, 47-13
  - users and groups, 17-21
- renumbering data blocks, 28-14
- .REP files, 35-13, 47-3
- Replace dialog box, 35-18
- replace operations, 20-11
- replacing
  - empty fields with values, 22-15
  - missing values with text, 36-25, 37-4
  - text in formulas, 26-19
  - text in report scripts, 35-18
  - text strings, 22-14
- replicated partitions
  - creating, 6-11, 6-12, 6-13, 16-4, 16-5, 16-6, 16-8, 16-11
  - defined, 6-10
  - disadvantages, 6-13
  - dynamically calculated values in, 29-26
  - example of, 6-35
  - guidelines for selecting, 6-15, 6-27
  - implementing security measures, 6-30
  - improving performance, 6-13
  - mapping members, 16-11, 16-17
    - by importing, 16-19
  - port usage, 6-15

- replicated partitions (*Continued*)
  - storage considerations, 15-12
  - updating data, 6-13, 16-5, 16-60
    - guidelines, 16-58
  - usage restrictions, 6-11
- replicating
  - See also* copying; duplicating
  - data files, 48-1
  - partial data sets, 6-10
  - problems with, 16-62
- Replication Properties dialog box, 16-5
- REPORT command, 35-5, 35-27
- Report Editor
  - creating scripts, 35-12
  - described, 35-5
  - editing scripts, 35-17, 35-18, 35-20
  - opening, 35-2
  - running report scripts, 35-21
  - specifying output destinations, 35-21, 35-22, 35-24
  - text editing commands, 35-11
- Report Extractor
  - associating members with dimensions, 36-4
  - described, 35-5
  - errors, 38-8
  - extracting data, 35-6
  - extraction order, 38-7
  - generating free-form reports, 35-28
  - ignored characters, 36-3
  - internal numerical comparisons, 38-5
- report files
  - creating, 35-24
  - naming, 35-25
- report formatting
  - commands, 35-9, 35-29
- report formatting commands
  - calculations, 36-16, 36-20
  - caution for usage, 36-48, 36-51
  - defined, 36-1
  - display options, 36-22, 36-26
  - listed, 36-9
  - nesting, 36-2
  - output, 36-15
  - report headings, 36-10, 36-13, 36-14, 36-15
  - types described, 36-8
- report generation functions, 36-52
- report output commands, 35-9
- Report Output Options dialog box
  - described, 35-21
  - opening, 39-9
- Report Script button (Application Manager), 7-8
- report script examples, 37-1
  - adding aliases, 37-20, 37-39
  - calculations, 37-32, 37-40, 37-44
  - changing data combinations, 37-14
  - creating asymmetric columns, 37-39
  - customizing page layouts, 37-23, 37-27
  - formatting layouts, 37-2, 37-17, 37-35, 37-37
  - grouping rows, 37-8
  - handling missing values, 37-4
  - narrowing member selection, 37-57
  - nesting columns, 37-6
  - ordering data, 37-55
  - restricting data retrieval, 37-54
  - sorting data values, 37-27, 37-32, 37-51
- report script files
  - adding scripts, 36-2
  - default location, 35-14
  - location of sample, 37-1
  - naming, 35-25
  - opening, 35-14, 35-16
- report scripts
  - See also* reports
  - adding comments, 36-3
  - adding conditions, 36-32, 36-45
    - examples, 37-54, 37-55, 37-57
  - adding members, 36-27, 37-57
  - adding variables, 36-33, 36-35, 36-36, 36-45
  - associating members with, 36-4
  - batch files and, 44-15
  - caution for aliases in, 36-13
  - caution for non-changing names in, 36-40
  - copying, 47-12
  - creating, 36-1, 36-2
    - basic techniques, 35-2, 35-12
    - in ESSCMD, 35-5
    - parts described, 35-9
  - creating with Application Manager, 7-8
  - currency conversions and, 43-29
  - defined, 7-4
  - defining page layouts, 36-3, 36-4, 36-7

report scripts (*Continued*)

- editing, 35-11, 35-17, 35-18, 35-20
  - exporting data with, 39-13
  - finding and replacing text in, 35-17, 35-18
  - formatting
    - calculated values, 36-16, 36-20
    - report layouts, 36-8, 36-10, 36-22
    - reports, 36-9
  - formatting calculated values, 36-16
  - inserting
    - equations in, 36-21
    - formulas in, 37-32
  - inserting UDAs, 36-37
  - loading, 35-15, 35-16
  - names with special characters, 8-16, 8-17
  - naming restrictions, 36-2
  - ordering data values, 36-45, 36-46, 37-55
  - resetting operators, 36-21
  - running
    - basic techniques, 35-21, 35-26, 35-27
    - examples, 36-5
    - in ESSCMD, 35-27
    - single line only, 35-27
    - with Application Manager, 35-13
  - running with Application Manager, 7-8
  - saving, 35-13
  - selecting members
    - with common attributes, 36-37
  - selecting members for column groups, 36-7
  - selecting precise member combinations, 36-32
  - sorting members, 36-44
  - suppressing shared member selection, 36-41
  - synchronizing outlines with, 38-8
  - unlocking, 46-40
  - using attribute members, 36-5
  - wildcards in, 36-38
- Report Viewer, 35-6
- Report Writer
- See also* report scripts
  - commands described, 35-9
  - creating text files, 39-1, 39-7
  - main components, 35-5
  - online help, 35-5
  - optimizing retrieval, 29-12, 38-2

## reporting objects (linked)

- assigning access levels, 12-4
  - changing member combinations, 12-9
  - checking structural integrity, 42-13
  - creating, 12-2
  - defined, 12-1
  - estimating disk space for, 15-13
  - format restrictions, 12-3
  - limiting size, 12-5
  - overview, 12-3, 12-4
  - removing from cells, 12-6
  - restructuring and, 40-20
  - retrieving, 12-3
  - storage management, 40-5
  - types supported, 12-2
  - viewing, 12-6, 12-8
- REPORTLINE command, 35-5, 35-27
- reports
- See also* time series reporting
  - ad hoc currency, 43-5
  - adding blank spaces, 36-26
  - adding calculated columns, 36-16, 36-20
  - adding headings, 35-7, 35-29, 36-4, 36-17
  - adding page breaks, 36-10, 36-23
  - adding titles, 36-24
  - adjusting column length, 36-13
  - associating with databases, 35-26
  - basic techniques, 35-1
  - building, 36-1, 36-7, 38-6
    - basic techniques, 35-2, 35-11
    - with API function calls, 36-52
  - calculating currency conversions, 36-51
  - changing
    - headings in columns, 36-8
    - layouts, 36-26
  - clearing values in calculated columns, 36-19, 36-20
  - creating two-dimensional, 39-13, 39-14
  - customizing page layouts, 36-6, 36-18
    - examples, 37-23, 37-27
  - defining multiple layouts, 37-36
  - designing, 5-4, 35-7, 35-10
  - developing free-form, 35-28
  - displaying member names, 36-43
  - dynamically calculated values and, 29-21

- reports (*Continued*)
  - eliminating data duplication, 36-41
  - improving readability, 11-1
  - numbering columns, 36-18
  - optimizing, 12-1, 36-29, 38-1
  - ordering output values, 36-47
  - printing, 35-22, 44-15
  - problems with formatting, 36-51
  - repeating column/row headings, 36-15, 36-18
  - replacing missing values, 36-25
  - restricting data retrieval, 36-46, 36-50
    - example, 37-54
  - retrieving data values, 35-6
    - setting maximum rows allowed, 36-50
    - with conditions, 36-45, 36-48
  - saving, 35-24, 35-25
  - setting up output destinations, 35-21, 35-22, 35-24
  - suppressing formatting in, 36-14, 36-23
  - terminating, 35-9
  - updating, 35-11
  - variance reporting examples, 5-31
- repositories, 5-3
- REPTKBYTESORTBUF setting, 38-5
- requests, 1-3, 45-12, 45-15
  - Dynamic Calculations and, 9-21
  - partitioned applications, 6-31
  - processing, 1-4, 45-13
- reserved names, 36-31
- reserved words, 8-16, 30-14
- RESETOTLCHANGETIME command, 16-56
- RESETSTATUS command, 44-11
- resources
  - checking usage, 46-6
  - minimizing, 4-13
  - optimizing network, 6-8, 6-12
  - partitioning databases and, 6-8, 6-9
- restarting system, 17-33
- restoring security settings, 17-2
- RESTRICT command
  - entering in report scripts, 36-46
  - NUMERICPRECISION parameter, 38-5
  - usage, 36-45
  - usage example, 37-54
- RESTRICT.REP, 37-54
- Restructure Database dialog box, 8-8, 40-27
- restructure operations, 40-15
  - improving performance, 40-18, 40-20
  - incremental, 40-19, 40-20
  - optimizing, 40-18, 40-19
  - outline changes impacting, 40-21, 40-27
- restructuring
  - attribute dimensions, 40-21
  - data blocks, 40-15, 40-18, 40-28
  - data files, 40-15, 40-27
  - databases, 40-14, 40-21
    - changing outlines and, 8-15, 8-20
    - dynamic calculations and, 29-23
    - immediately, 40-19, 40-22
    - Intelligent Calculation and, 34-3, 34-19
    - process described, 40-16, 40-20
  - indexes, 40-17
  - linked reporting objects and, 40-20
  - outlines, 8-8
    - prerequisites for, 17-38
  - partitioned databases, 6-19, 40-21
  - recovery and, 40-17, 48-7
- restructuring conflicts, 40-17
- Retrieval Buffer Size option, 38-3
- retrieval buffers, 29-12, 38-2, 38-4
- retrieval factor, 29-10
  - displaying, 29-10
- Retrieval Sort Buffer Size option, 38-4
- Retrieval Wizard, 39-7
- retrieving
  - cross-database values, 26-45
  - data, xliv
  - data blocks, 4-5, 4-11, 4-12
    - size and density, 4-8
  - data values for reports, 35-6
    - placing restrictions on, 36-46, 36-50, 37-54
    - setting maximum rows allowed, 36-50
    - with conditions, 36-45, 36-48, 37-54, 37-55
  - data values from remote databases, 6-15, 29-14, 29-25
  - Dynamic Time Series members, 30-13
  - dynamically calculated values, 29-4, 29-10, 29-16, 29-21
  - linked reporting objects, 12-3
  - member combinations, 4-7
  - partition type, 6-6
  - period-to-date values, 30-14

- retrieving (*Continued*)
  - specific values, 4-6, 26-44
  - unique values, 4-5
  - values for sparse dimensions, 4-5
- reversing data values, 22-27
- RLE data compression, 15-10, 40-9, 41-27
  - overview, 40-11
  - specifying, 41-27
- rollbacks, 42-11, 45-3, 45-9
  - outlines, 46-35
- rolling average values, 27-3
- rolling back transactions. *See* recovery
- roll-ups
  - See also* consolidation
  - building multiple, 13-44
  - examples, 13-32, 13-33, 13-37
  - implementing, 5-26
  - maximum number in records, 13-38, 13-40
  - member consolidation and, 9-16
  - multiple data sources, 13-44
  - optimizing, 33-6
  - ordering in rules files, 14-17
  - setting for time balance properties, 9-4, 9-5
  - shared members, 13-4, 13-37
- root member (defined), 3-6
- @ROUND function, 26-38
- rounding, 24-12, 26-38
- routine operations, 2-4
  - privileges and, 17-3
- routines, 5-31, 26-4
- row calculation commands, 36-20
- ROW command
  - entering in report scripts, 36-4
  - usage example, 37-13
- row headings, 35-29
  - adding to reports, 36-4
  - defined, 35-8
  - names truncated, 36-13
  - repeating, 36-15
  - suppressing, 36-14
- ROWAVG.REP, 37-50
- ROWCALC1.REP, 37-44
- ROWCALC2.REP, 37-46
- ROWGROUP.REP, 37-13
- ROWREPEAT command
  - entering in report scripts, 39-13
  - usage, 36-15
  - usage examples, 37-38
- rows
  - See also* records
  - attributes in report scripts, 36-5
  - calculating totals, 36-20
  - calculation sample reports, 37-44, 37-45, 37-47
  - formatting
    - empty values in, 36-14
    - missing values in, 36-14
    - with different member names, 37-39
  - grouping in reports, 37-8
  - in data sources, 20-1
  - restrictions on retrieval, 36-50
  - setting qualifications for retrieval, 36-46, 36-48, 36-50
  - setting transactions levels, 41-16
  - switching with columns, 37-8
- .RPT files, 35-25
- @RSIBLINGS function, 26-37
- .RUL files, 47-3
  - See also* rules files
- rules
  - creating aliases, 11-2
  - creating shared members, 9-23
  - data entry in member fields, 20-5
  - data load
    - creating, 20-9
    - when to use, 20-8
  - data loads, 7-3
  - defining attribute dimensions, 9-13
  - defining attributes, 13-18
  - defining dimension type, 9-2, 9-3
  - dimension builds, 7-3
  - formatting data sources, 20-12, 20-14, 20-15, 20-17, 20-18
  - formatting free-form data sources, 20-19, 20-22
  - replicated partitions, 6-11
  - replicating
    - data, 6-13
  - transparent partitions, 6-16, 6-20
  - UDAs, 9-25

- rules files
    - adding members with, 13-13, 13-15
    - associating
      - with data sources, 23-9
      - with outlines, 14-6, 14-9, 14-14, 21-22
    - associating aliases with attributes, 13-26
    - attribute associations, 13-20
    - bottom-up ordering, 13-9, 13-43
    - building dimensions, 14-1, 14-2, 14-26, 23-10
    - building shared members, 13-32
      - different generation, 13-38, 13-39
      - same generation, 13-34, 13-35, 13-36
      - with branches, 13-41, 13-42
    - changing field names, 22-16
    - copying, 47-12
    - creating, 7-8, 13-2
      - process overview, 21-1
    - creating new fields, 22-7
    - cross-platform compatibility, 47-16
    - for dimension builds, 13-1
    - generation references and, 13-5, 13-6, 13-34
    - header records and, 14-23, 21-11
    - initializing, 14-31
    - invalid, 14-22
    - level references and, 13-9, 13-35, 13-38, 13-41
    - loading, 14-28, 23-9
      - prerequisites, 23-2
    - logging load errors, 23-12
    - manipulating fields in, 22-1
    - naming, 21-25
    - opening, 21-3
    - optimizing usage, 21-13
    - options for setting up, 13-3
    - parent/child builds, 13-12, 13-36, 13-39, 13-42
    - position of fields, 13-22, 14-17
    - replacing text strings, 22-14
    - restrictions for entering field types, 14-17
    - saving, 21-21, 21-24, 21-25
    - setting global options, 14-20
    - specifying field types, 14-12
    - SQL data sources and, 20-3, 21-8
    - top-down ordering, 13-6
    - usage overview, 13-2, 20-8, 20-10, 21-11
    - validating, 14-21, 21-21, 21-23
      - troubleshooting problems with, 24-6
      - with blank fields, 20-17
  - Run button (Application Manager), 7-8
  - RUNCALC command, 31-34
  - Run-Length Encoding. *See* RLE data compression
  - running
    - batch files, 44-9, 44-10
    - calc scripts, 31-7, 31-24, 31-31, 31-50
      - aborted transactions and, 42-12
      - on partitioned applications, 31-57
    - calc scripts with Application Manager, 7-8
    - currency conversions, 43-26, 43-27, 43-30
    - ESSCMD, 44-3, 44-5
      - caution for pausing systems while, 44-8
      - in interactive mode, 44-2
    - multiple report script files, 35-27
    - report scripts
      - basic techniques, 35-21, 35-26, 35-27
      - examples, 36-5
      - in ESSCMD, 35-27
      - single line only, 35-27
      - with Application Manager, 35-13
    - report scripts with Application Manager, 7-8
  - RUNREPT command, 35-27
  - run-time information, 46-14
  - Run-time page (Database Information), 15-18, 46-14
- ## S
- salary databases, 5-15, 26-31
  - sales, 27-5
    - allocating percentages for, 26-47
    - estimating profits, 32-17
    - period-to-date values, 30-7
    - reporting
      - sample for market selection, 37-6
      - sample for percentages, 37-32
      - samples for actual, 37-2, 37-44
    - year-to-date and rolling averages, 27-3
  - Sampeast application, xliv
    - partitioning examples, 6-34
    - replicated partitions in, 6-10, 16-4
    - transparent partitions in, 16-20
  - Sampeast East database
    - provided with Hyperion Essbase, xliv
  - Sample application, xliv
    - creating simple reports, 35-2
    - currency conversion databases, 43-3

## Sample Basic database

- batch mode processes, 44-12
  - consolidation example, 3-4
  - creating calc scripts, 31-3
  - creating formulas for outlines, 26-8
  - creating outlines for, 8-1
  - defining calculations for, 28-5, 28-6, 28-7
    - actual values, 31-2
    - allocating values, 32-7, 32-13
    - allocating values across one dimension, 32-9
    - data subsets, 32-3
    - forecasting future values, 32-21
    - forward references and, 28-10
    - increasing budgets, 31-6
    - percentage of variance, 32-2
    - product and market share, 32-6
    - sales and profit, 32-17
    - shared members in, 28-26
    - specific cells, 28-15, 28-16, 28-18, 28-20
  - dense dimensions in, 4-2, 4-5, 24-8
  - dynamically calculating data, 29-14
  - generating reports, 36-5
  - header and mapping information in, 20-13
  - Intelligent Calculations on, 34-9, 34-10, 34-12
  - loading new budget values, 32-5
  - optimal dimension configurations, 4-9
  - optimizing calculations in, 33-10, 33-11, 33-24
  - partitioning examples, 6-33, 6-36, 6-38, 16-31
  - provided with Hyperion Essbase, xliv
  - report calculations in, 36-21
  - report formatting examples, 36-10, 36-12, 36-17
  - sample reports, 37-1
  - selecting members for reports, 36-30, 36-40
  - sorting example, 36-49
  - sparse dimensions in, 4-2, 4-4, 24-8
  - two-pass calculations in, 29-17
- sample databases, xliv
- SAMPLE directory, 35-14
- sample ESSCMD script files, 44-12
- sample reports, 37-1
- adding aliases, 37-20, 37-39
  - changing data combinations, 37-14
  - creating asymmetric columns, 37-39
  - customizing page layouts, 37-23, 37-27
  - formatting layouts, 37-2, 37-17, 37-35, 37-37

sample reports (*Continued*)

- grouping rows, 37-8
  - handling missing values, 37-4
  - narrowing member selection, 37-57
  - nesting columns, 37-6
  - ordering data, 37-55
  - restricting data retrieval, 37-54
  - sorting data values, 37-27, 37-32, 37-51
  - using attributes, 37-58, 37-60
  - using attributes in member selection, 37-58
- sample script files, 44-12
- Samppart application, xliv
- partitioning examples, 6-34
  - replicated partitions in, 6-10, 16-4
  - transparent partitions in, 16-20
- Samppart Company database
- provided with Hyperion Essbase, xliv
  - @SANCESTVAL function, 26-44
- Save As command, 47-19
- Save Client Object dialog box
- saving calc scripts, 31-27, 31-29
- Save Server Object dialog box
- copying outlines, 8-5
  - saving calc scripts, 31-25, 31-27, 31-29
  - saving rules files, 21-24, 21-25
- SAVEANDOUTPUT command, 36-20
- SAVEROW command
- restrictions, 36-51
  - usage, 36-20
  - usage example, 37-33
- saving
- attachments, 12-3
  - calc scripts, 31-5, 31-24, 31-25
    - on client workstations, 31-27, 31-29
  - filters, 18-2
  - formulas, 26-14
  - outline files, 39-6
  - outlines, 8-5, 8-7, 8-12, 40-27, 40-28, 40-29
    - caution for, 8-8
  - output files, 39-9
  - partition definitions, 16-40
  - report scripts, 35-13
  - reports, 35-24, 35-25
  - rules files, 21-21, 21-24, 21-25
  - security profiles, 17-13, 17-18, 17-19

- scalability, 6-8
- scaling data values, 22-25
- scope
  - checklist for analyzing, 5-17
  - determining, 5-4
  - Hyperion Essbase kernel settings, 41-4
- .SCR files, 44-9, 47-3
- script files. *See* ESSCMD script files
- scripts. *See* calc scripts; report scripts
- searches
  - case-sensitive, 22-15, 26-19, 26-23, 31-42, 35-17
  - large indexes and, 4-11
  - members in Calc Script Editor, 31-46
  - members in Formula Editor, 26-22
  - returning specific values, 4-6, 26-44
  - sequential, 4-6
  - text
    - in calc scripts, 31-42
    - in formulas, 26-19
    - in report scripts, 35-17, 35-18
  - viewing information about index, 46-14
- season-to-date calculations, 30-8
- .SEC files, 17-2, 47-4
- secondary fields, 13-7, 13-10
- secondary roll-ups, 13-38, 13-40, 14-17
- security, 18-1, 35-10
  - See also* access; filters; privileges
  - access levels listed, 17-25, 17-30, 18-2
  - backup files, 17-2
  - changing for users and groups, 17-12, 17-21
  - checking information about, 46-2
  - implementing, 5-17, 25-12
    - for users and groups, 17-5, 17-7, 17-9, 17-16
    - globally, 17-29, 17-31, 17-34, 17-35
    - guidelines for, 2-3
    - system server, 17-36, 17-37, 17-39
  - information file, 17-2
  - layers defined, 17-1
  - linked reporting objects, 12-4
  - maintaining information for, 7-4
  - managing, 17-1, 17-4, 17-29, 17-36
    - for large number of users, 17-27
  - modifying user access settings, 17-22
  - overriding, 17-33
- security (*Continued*)
  - planning for, 20-24
  - profiles
    - copying, 17-16, 17-18
    - creating, 17-9
    - deleting users and groups, 17-19
    - editing, 17-12
    - saving, 17-13, 17-18, 17-19
  - sample solutions, 19-1, 19-2, 19-3, 19-4, 19-5
  - saving information to ASCII files, 45-5, 45-12
  - setting up for partitioned databases, 6-30, 6-31, 6-32
- security file
  - backup of, 17-2
  - contents of, 17-2
  - cross-platform compatibility, 47-16
  - filter storage, 18-2
  - restoring, 17-2
- Security System, 17-1
  - See also* security
- security types
  - Create/Delete Users/Groups, 17-6
  - defined, 17-5
  - ordinary, 17-6
  - Supervisor, 17-5
- .SEL files, 47-3
- SELECT command, 44-7, 45-6, 45-8
- select criteria. *See* selection criteria
- Select Database dialog box, 35-26
- Select Record dialog box, 21-15
- Select Server, Application and Database dialog box, 21-8
- select statements. *See* SQL databases
- selecting
  - alias tables, 26-25, 31-48
  - applications for loading, 7-7
    - with ESSCMD, 44-7
  - build methods, 13-4, 14-9
  - calc script files, 31-21, 31-22
  - data sources, 6-9, 16-4, 21-1, 23-3, 23-5
    - using Windows, 23-7
  - data targets, 16-4
  - data to partition, 6-3, 6-8, 6-9
  - databases for loading, 7-7
    - with ESSCMD, 44-7

selecting (*Continued*)

- dimension type, 4-8, 4-11, 4-13, 4-14
    - guidelines for, 5-7
  - fields, 22-2
  - members for dynamic calculations, 29-13
    - guidelines for, 29-13, 29-15
  - members for report scripts, 36-27, 36-29, 36-39
    - from Dynamic Time Series, 36-31
    - with ATTRIBUTE command, 36-35
    - with Boolean operators, 36-32
    - with conditions, 37-57
    - with substitution variables, 36-33
    - with TODATE command, 36-36
    - with user-defined attributes, 36-37
    - with wildcards, 36-38
    - with WITHATTR command, 36-36
  - members, for column groupings, 36-7
  - multiple dimensions and members, 37-36
  - multiple files, 23-7, 23-8
  - multiple members, 8-22
  - partition type, 6-10, 6-15, 6-20, 6-26, 6-27
    - in Partition Wizard, 16-3, 16-5, 16-21, 16-31
  - records, 21-15, 21-16, 21-18
  - report script files, 35-15, 35-16
  - SQL data sources, 23-4
  - values for dynamic calculations, 29-6, 29-15
    - guidelines for, 29-7, 29-8
- selection criteria, 21-15
- defining multiple for single field, 21-16
  - defining on multiple fields, 21-18
  - specifying in rules file, 20-12
- semantic errors, 26-25, 31-48
- semicolons (;)
- in application and database names, 7-12
  - in calc scripts, 31-15, 31-17, 31-36, 31-40
  - in ESSCMD syntax, 44-2
  - in formulas, 26-6, 26-7, 26-16
  - in names in scripts and formulas, 8-17, 36-3
- separators
- See also* file delimiters
  - commas in numbers, 20-4
  - member lists, 26-35
  - report script commands, 36-2, 36-15
- sequential searches, 4-6

- server, 1-4, 7-2
- application startup, 7-3
  - caution for improper shutdown, 17-33
  - caution for rebooting, 17-33
  - changing system password for, 45-5, 45-11
  - client requests and, 45-12, 45-13, 45-15
  - client/server model described, 1-4
  - client-server model described, 1-4, 1-5
  - communicating with clients, 45-1
  - connecting to. *See* connections
  - crashes, recovering from, 24-5, 48-5, 48-6
    - suggested procedures, 48-7
  - cross-platform compatibility, 47-15
  - disconnecting users, 17-35, 17-38
  - displaying
    - access settings, 17-27, 17-28
    - active groups, 17-27
    - current users, 17-8, 45-4, 45-10
  - displaying version number, 45-5, 45-12
  - functionality described, 1-3
  - getting information about, 46-3, 46-31
  - installing, xliii
  - interruptions, 46-26
  - loading data from, 24-11
  - logging errors, 46-26
  - logging out of, 17-33, 17-39, 45-4, 45-10
  - moving data sources to, 24-11
  - partitioning databases across multiple, 6-8
  - redefining information, 47-19
  - restrictions, 23-7
  - securing, 17-36, 17-37, 17-39
  - setting internal buffer, 38-2
  - shutting down, 45-2, 45-3
  - starting, 45-2
  - system supervisor defined, 17-5
  - unavailable, 24-1, 24-4
  - viewing log file, 46-31
  - writing calc scripts to, 31-25
- Server Agent, 1-4, 45-1
- See also* server console
  - accessing, 45-2
  - activity recording, 45-4, 45-16
  - commands listed, 45-4
  - displaying available commands, 45-5

- Server Agent (*Continued*)
  - handling requests, 45-13
  - monitoring applications, 45-16
  - overview client-server communications, 45-12, 45-14
  - setting number of threads, 45-15
- server applications. *See* client-server applications
- server console, 45-2
  - exception occurred messages in, 46-26
  - overview, 45-4
  - remote shutdowns, 45-3
  - shutting down server, 45-2, 45-3
  - starting applications, 45-5, 45-6
  - starting databases, 45-8
  - starting Essbase server, 45-2
  - stopping applications, 45-5, 45-6
    - multiple simultaneously, 45-12
  - stopping databases, 45-8, 45-9
- server errors, 46-41
- server event log files, 46-31
  - deleting, 46-32
  - viewing contents, 46-31
- Server Information dialog box, 46-3
- server interface, 7-7
- Server Settings dialog box
  - auto logoff, 17-40
  - password management, 17-40
- SERVERTHREADS setting, 45-15
- SET AGGMISSG command
  - and allocating costs, 32-8
  - and member combinations, 31-14
  - described, 31-13, 33-36
- SET CACHE command, 31-13
- SET CALCHASHTBL command, 31-13, 33-22
  - performance considerations and, 33-4
- SET CLEARUPDATESTATUS command, 31-13
  - calculating subsets with, 34-4
  - concurrent calculations and, 34-12
  - Intelligent Calculation and, 31-54, 34-3, 34-6, 34-7, 34-9, 34-11
  - multi-pass calculations, 34-14, 34-15, 34-16
  - parameters listed, 34-7
  - two-pass calculations, 33-32, 33-33
- Set Current Alias Table dialog box, 11-8
- Set Database Note dialog box, 7-13
- Set Default Calc dialog box, 25-8
- SET FRMLBOTTOMUP command, 31-13, 33-8
- SET LOCKBLOCK command, 31-14, 33-23
- SET MSG command, 31-13, 31-14
- SET MSG DETAIL command, 33-4
- SET MSG SUMMARY command, 33-4, 33-5
  - usage example, 33-21
- SET NOTICE command, 31-13, 33-4, 33-5
- SET UPDATECALC command, 31-13, 34-5
  - usage example, 33-32
- SET UPDATECALC OFF command, 32-3
- SET UPTOLOCAL command, 31-14
- SETALIAS command, 11-8
- SETAPPSTATE command, 15-13
- SETCENTER command, 36-9
- SETDBSTATE command, 41-3, 41-7, 46-9
  - allocating storage, 40-13
  - changing data compression, 40-10
  - precedence, 41-4
  - running in batch mode, 41-7
  - setting cache size, 41-9, 41-12
  - setting index page size, 41-15
  - specifying data compression, 41-28
- SETDBSTATEITEM command, 41-3, 41-7
  - aggregating missing values, 33-2, 33-9, 33-27, 33-36
  - allocating storage, 40-13
  - changing data compression, 40-10
  - increasing retrieval buffer, 29-13
  - precedence, 41-4
  - running in batch mode, 41-7
  - scope of storage settings, 41-4
  - setting
    - cache size, 41-9, 41-12
    - index page size, 41-15
    - retrieval buffer size, 38-3
    - retrieval sort buffer, 38-5
    - transaction isolation levels, 41-18
  - specifying
    - data compression, 41-28
    - disk volumes, 41-24, 41-26
  - viewing database information, 46-9

- SETDEFAULTCALC command, 25-8
- SETDEFAULTCALCFILE command, 25-8
- SETPASSWORD command, 17-13, 45-11
- SETROWOP command
  - entering in report scripts, 36-21
  - usage, 36-20
- setting
  - See also* assigning; defining; applying
  - aggregate missing values, 33-37
  - attribute types, 9-14
  - cache size, 15-16, 33-14, 33-18
    - first-time calculations, 33-21
    - with Application Manager, 41-8, 41-10
    - with ESSCMD, 41-9, 41-10
    - with Hyperion Essbase kernel, 41-8, 41-9
  - conditions in formulas, 26-29, 26-31
  - configurable variables, 38-2
  - consolidation properties, 9-16, 9-18
  - default calculations, 25-7, 33-27, 34-3, 34-5
  - dimension and member properties, 9-1
  - file delimiters, 20-6, 20-18, 21-10
  - fonts and typefaces, 8-26
  - global options, 14-20, 31-13
  - index page size, 41-14, 41-15
  - internal server buffer, 38-2
  - member consolidation properties, 14-24
  - passwords and user names
    - new users, 17-10, 17-17
    - overview, 17-39
    - partitioned databases, 16-6
  - properties dynamically, 14-24
  - report output destinations, 35-21, 35-22, 35-24
  - retrieval buffer size, 29-12, 38-2
    - for data sorts, 38-4
  - transaction isolation levels, 41-15, 41-17, 41-19
    - with ESSCMD, 41-7, 41-18
- shaded cells, 3-9
- shared areas. *See* partition areas
- shared library files, 47-4
- shared locks. *See* Read locks
- shared member property, 5-21, 9-20
- shared members
  - adding to outlines, 13-32, 13-42, 13-43
    - caution for placing, 9-23
  - affect on consolidation paths, 5-28
- shared members (*Continued*)
  - at different generations
    - building dynamically, 13-37, 13-38, 13-39
  - at same generation, 13-33, 13-34
    - building dynamically, 13-35, 13-36
  - building dynamically, 13-32, 13-33, 13-37, 13-40
  - calculating, 28-26, 29-5
  - caution for sorting, 8-21
  - creating, 9-23, 13-40, 13-42, 13-44
    - for different generations, 13-38, 13-39
    - for same generation, 13-33, 13-35, 13-36
    - guidelines for, 9-23, 13-4
    - with Outline Editor, 9-25
    - with rules files, 13-32
  - described, 9-23
  - design approach for attributes, 10-13
  - different generations, 13-37
  - getting, 26-44
  - guidelines, 5-15, 9-23
  - implicit, 5-22, 9-23, 24-3
  - linked reporting objects and, 12-3
  - parent/child
    - most versatile build method, 13-42
  - partitioned applications and, 16-25, 16-30
  - properties, 9-25
  - relationship implied, 9-23
  - reorganizing, 14-10
  - rolling up, 13-32
  - same generation, 13-33
  - sample rules files
    - generation references, 13-34
    - level references, 13-35, 13-38, 13-41
    - parent/child references, 13-36, 13-39, 13-42
  - storing, 15-4
  - suppressing for report generation, 36-41
  - with branches
    - building dynamically, 13-40, 13-41, 13-42
    - with outline synchronization, 16-53
- shared partition areas, 16-48
- shared roll-ups, 13-44
- sharing data, 6-4, 6-6
  - across multiple sites, 6-9, 13-44
  - in partitioned databases, 16-8, 16-22, 16-33
  - never allow property, 9-20, 9-24, 14-25
  - not allowing, 5-22

- sharing members. *See* shared members
- sheets. *See* Spreadsheet Add-in; spreadsheets
- shell scripts (UNIX), 44-9
- SHGENREF.RUL, 13-34
- SHGENREF.TXT, 13-34
- @SHIFT function, 26-44
- SHLEV.RUL, 13-35
- SHLEV.TXT, 13-35
- Show Warnings option, 35-22, 35-25
- shutdown (caution for improper), 17-33
- SHUTDOWNSERVER command, 45-3
- shutting down server, 45-2, 45-3
- @SIBLINGS function, 26-37
- siblings
  - adding as members, 13-13, 13-15
  - calculation order in outlines, 9-15
  - checking for, 26-30
  - consolidation properties and, 9-16
  - defined, 3-6
  - dimensions as, 8-18
  - getting, 26-37
  - members as, 8-19, 8-22
  - rolling up, 13-37
- SIBLOW.RUL, 13-15
- SIBLOW.TXT, 13-15
- SIBPAR.TXT, 13-17
- SIBSTR.RUL, 13-13
- SIBSTR.TXT, 13-13
- Sign Flip option, 22-28
- Simple application, 3-12
  - changing makeup, 3-16, 3-18, 3-19
  - organizing data, 3-13
- simple formulas, 33-6, 33-13
- simple interest, 26-45
- single quotation marks (')
  - in application and database names, 7-12
  - in dimension and member names, 8-15
- single-server applications
  - adding dimensions, 5-6, 5-22
  - analyzing data, 5-4, 5-6
  - analyzing database scope, 5-11, 5-17
  - creating outlines, 5-17
  - defining calculations, 5-25, 5-26, 5-27, 5-28, 5-29, 5-31
  - designing, 5-1, 5-4
  - single-server applications (*Continued*)
    - identifying data sources, 5-4
    - implementing security, 5-17
    - partitioned vs., 5-1
    - planning process, 5-3, 5-4
  - size
    - array variables, 31-13
    - change log files, 46-39
    - data blocks, 4-8, 15-5, 15-6, 33-2
      - controlling, 41-29
    - determining cache, 15-14, 15-15
    - estimating database, 3-14, 15-1, 15-8, 15-9
    - factors determining database, 3-14
    - field and optimal loads, 24-12
    - getting information on cache, 46-11
    - index page, 41-14, 41-15
    - linked files, 12-4, 12-5
    - minimizing for linked objects, 12-5
    - optimizing index, 4-11, 4-12, 4-13
    - partitioned databases, 15-12
    - planning for optimal, 5-4
    - reducing database, 6-13
    - setting cache, 15-16, 33-14, 33-18
      - first-time calculations, 33-21
      - with Application Manager, 41-8, 41-10
      - with ESSCMD, 41-9, 41-10
      - with Hyperion Essbase kernel, 41-8, 41-9
    - setting limits for linked objects, 15-13
    - setting maximum file, 41-20
    - setting retrieval buffer, 29-12, 38-2
      - for data sorts, 38-4
- SKIP command
  - usage, 36-26
  - usage example, 37-18
- Skip Missing property, 30-5
- skip properties, 9-6
- Skip Zeros property, 30-5
- SKIPONDIMENSION command, 36-26
- skipping
  - #MISSING and zero values, 9-7, 30-5
  - fields when mapping, 22-2, 22-3, 22-4
  - lines in rules files, 20-11, 21-14
  - multiple fields in data load, 21-18
  - records in data load, 21-16
  - specific fields in data load, 21-17

- .SL files, 47-4
- slashes (/)
  - as codes in data source, 14-25
  - in names in scripts and formulas, 8-17, 36-3
- slicing, 3-11
- @SLN function, 26-46
- smoothing data, 26-42
- SMP (symmetric multiprocessing), 1-3, 45-15
- .SO files, 47-4
- software version, 45-5, 45-12
- Solaris servers. *See* UNIX platforms
- Sort Descending command, 8-21
- sort order
  - defining, 36-46
  - members in outline, 8-21
  - output, 36-47
- SORTALTNAMES command, 36-44
- SORTASC command, 36-44
- SORTDESC command, 36-44
- SORTGEN command, 36-44
- sorting
  - data for reports, 35-7, 36-45, 36-46
    - examples, 37-27, 37-32, 37-51
  - data with missing values, 36-49
  - dimensions and members, 8-21, 14-11
  - members in reports, 36-44
  - records to optimize data loading, 24-8
- sorting commands, 36-44, 36-46
- SORTLEVEL command
  - usage, 36-44
  - usage example, 37-31
- SORTMBRNames command
  - precedence in sorts, 36-46
  - usage, 36-45
- SORTNONE command, 36-45
- source data
  - changing case, 22-16
- source outline (defined), 16-47
- space. *See* white space
- spaces
  - in application and database names, 7-12
  - in dimension and member names, 8-16
- @SPARENTVAL function, 26-45
- SPARSE command, 29-21
- sparse dimensions, 4-2
  - See also* dense dimensions; dimensions
  - calculating values in, 28-23, 31-56, 33-8, 33-22
  - defined, 4-2
  - defining member combinations for, 28-11, 28-25
  - Dynamic Calc, 5-23, 10-14
  - dynamically calculating, 29-7, 29-13, 29-14, 29-16, 29-23
  - formulas and, 33-8
  - grouping member combinations, 24-8
  - implications for restructuring and, 40-15, 40-18
  - Intelligent Calculation and, 34-11
  - location in outline, 5-23, 10-14
  - marked as clean, 31-50
  - optimizing, 24-8, 24-10, 33-3
  - partitioning, 6-14
  - recalculating values in, 31-50, 34-11
  - reporting on, 38-7
  - returning values for, 4-5
  - selecting, 4-8
  - selection scenarios, 4-11, 4-13, 4-14
  - setting, 14-7
  - storing, 8-9
  - storing member combinations, 28-3
  - unique member combinations, 4-4
- sparse restructure, 40-15
- speed up. *See* optimizing
- @SPLINE function, 26-43
- spline, calculating, 26-43
- split dimensions, 5-14
- Split Field dialog box, 22-9
- splitting
  - databases, 5-16, 6-2
  - fields, 22-9
- Spreadsheet Add-in, 1-2, 45-13
  - ad hoc currency reporting, 43-5
  - as client interface, 1-5
  - getting Dynamic Time Series members, 30-13
  - linked partitions and, 6-23, 6-25
  - linked reporting objects and, 12-2, 12-6
  - maintaining data integrity, 17-37
  - optimizing retrieval, 29-12, 38-2
  - running calc scripts, 31-31
  - specifying latest time period, 30-15
  - viewing database information, 7-13
- spreadsheet files, 47-1, 47-3



- storage (*Continued*)
  - data values, 9-20, 9-23
  - deallocation and, 41-22, 41-26
  - default properties, 5-21
  - dimensions, 8-9
  - dynamically calculated values, 29-2, 29-3
    - with attributes, 10-31
  - fine-tuning, 2-4
  - index files, 40-12, 41-20
  - inefficient, 4-16
  - internal structures optimizing, 4-4
  - linked reporting objects, 12-2, 12-3, 15-13
  - local, 6-10
  - multiple applications, 7-2
  - optimizing, 6-8
  - partitioned databases, 6-8, 6-9, 15-12
    - sparse member combinations and, 28-3
    - with replicated partitions, 6-13
    - with transparent partitions, 6-18
  - planning for, 5-5
  - restructuring and, 40-14
  - server configurations, 1-4
  - shared members, 15-4
  - temporary variables, 31-12
  - trace files, 46-34
  - viewing information about, 4-8
- Storage page (Database Information), 46-11
- Storage page (Database Settings), 41-6
  - setting
    - cache size, 41-8, 41-10, 41-11
    - index page size, 41-14
  - specifying
    - data compression, 41-27
    - disk volume, 41-22
- storage settings, 41-2, 41-3
  - scope, 41-4
- store data property, 5-21
- store property
  - description, 9-20
- Stores, 40-5
- .STR files, 8-12, 47-3
- strings
  - See also* characters
  - adding fields by matching, 22-4
  - adding members by matching, 13-13
  - as tokens, 22-2
- strings (*Continued*)
  - calc scripts as, 31-24
  - dummy, 22-15
  - options for loading, 21-16, 21-17
  - preceded by &, 26-48
  - replacing for data loads, 22-14
- subexpressions, 37-58
- subsets of data
  - calculating, 31-12, 31-54, 31-55
    - example, 32-3
    - with Intelligent Calculation, 34-4
  - clearing, 31-53
  - copying, 31-54
  - copying to Personal Essbase, 39-2
    - prerequisites, 39-7
  - loading, 34-1, 39-9
- substitution variables, 7-14
  - adding to report scripts, 36-33, 36-35, 36-36
  - creating, 7-15
    - guidelines for, 7-15
  - deleting, 7-17
  - formulas and, 26-48
  - free-form reports and, 35-30
  - inserting in calc scripts, 31-12, 31-52
  - updating, 7-18
  - usage overview, 26-48
- Substitution Variables dialog box, 7-16, 7-17
- subtotals, 36-16, 36-20
- subtracting
  - values from existing values, 22-22
- subtraction
  - consolidation codes in data source, 14-25
  - prerequisite for, 22-23
  - setting member consolidation properties, 9-18
- suffixes
  - adding to fields, 22-20
  - assignment sequence in data build, 20-11
  - attribute member names, 9-13, 10-16, 10-17
  - member names, 8-17
- @SUM function, 26-38
- Sum member, Attribute Calculations dimension, 10-20, 10-32
- summaries, 35-1
  - reporting example, 37-45
- summary information, 36-24
- Summary page (Partition Wizard), 16-40

- @SUMRANGE function, 26-43
- sums, 26-38, 36-16
- SUPALL command, 36-14
- SUPBRACKETS command, 36-23
  - overriding, 36-26
  - usage example, 37-18
- SUPCOLHEADING command, 36-14
- SUPCOMMAS command, 36-23
- SUPCURHEADING command, 36-14
- SUPEMPTYROWS command
  - entering in report scripts, 36-50
  - usage, 36-14
- Supervisor privilege, 17-5
- SUPEUROPEAN command, 36-23
- SUPFEED command
  - usage, 36-23
  - usage example, 37-38
- SUPFORMATS command, 36-23
- SUPHEADING command, 36-14
- SUPMASK command, 36-23
- SUPMISSINGROWS command
  - usage, 36-14
  - usage example, 37-7
- SUPNAMES command, 36-14
- SUPOUTPUT command, 36-14
- SUPPAGEHEADING command
  - usage, 36-14
  - usage example, 37-31
- SUPPMISSING command, 36-50
- supported platforms, 1-5
- suppressing report formatting, 36-14, 36-23
- SUPPRESSMISSING command, 37-56
- SUPSHARE command, 36-41
- SUPSHAREOFF command, 36-41
- SUPZEROROWS command, 36-23
- SUPZEROS command, 36-50
- swap space, 46-7
- swapping devices, 46-6
- Sybase SQL Server. *See* SQL databases
- @SYD function, 26-46
- SYM command
  - entering in report scripts, 36-8
  - usage, 36-4
  - usage example, 37-21
- symmetric columns, 20-22
- symmetric multiprocessing (SMP), 1-3, 45-15
- symmetric reports
  - asymmetric reports vs., 38-6
  - calculated columns in, 36-17
  - changing column headings, 36-8
  - creating, 36-7, 38-6
  - defined, 36-7
  - formatting, 36-4
  - outputting column groups, 37-22
  - overriding column groupings, 36-8
- Synchronize Outline dialog box, 16-50
- synchronizing
  - data, 6-2, 6-8, 6-36
    - partitioning and, 40-21
  - outlines, 6-2, 6-25
    - process summarized, 16-47, 16-48
    - warning for not applying changes, 16-49
    - with Application Manager, 16-50
    - with Partition Wizard, 16-4
  - outlines with report scripts, 38-8
- syntax
  - See also* formats
  - calc scripts, 31-15
  - calculation commands, 31-11
  - checking in Calc Script Editor, 31-5, 31-48, 31-49
  - checking in Formula Editor, 26-25, 26-26
  - commands and functions, xliii
  - comments, 9-27
  - ESSCMD, 44-2
  - formulas, guidelines for, 26-6
  - report scripts, 36-2
    - member selection, 36-29, 36-30
    - substitution variables, 36-34
    - user-defined attributes, 36-38
    - with wildcards, 36-39
  - verifying for filters, 18-6
- syntax errors
  - finding in calc scripts, 31-48
  - finding in formulas, 26-25
  - formulas and dynamically calculated members, 29-8
    - stepping through, 26-26
- system administrators. *See* administrators
- system errors, 46-26, 46-41
  - log file locations and names, 46-27
- system failures. *See* failures; recovery

System Info page (Server Information), 46-6  
 system information, 46-3, 46-6  
 system password, 45-5  
   changing, 45-11  
 system security. *See* security  
 System Supervisor, 17-5

## T

T, two-pass calc code in data source, 14-25

tab

- as column separator, 36-15
- as command separator, 36-2
- as file delimiter, 20-6
- in names, 8-15

TABDELIMIT command

- entering in report scripts, 39-8
- usage, 36-15

tab-delimited formats, 36-15

tables. *See* alias tables; databases

tags

- See also* properties
- assigning to members, 5-21
- display options, 8-25
- usage examples, 5-29

target outline, 16-47

- See also* targets

targets

- accessing data, 6-10, 6-18
- calculations and, 6-14, 6-21
- changing data in, 6-11
- changing outlines, 16-48
- copying from, 6-10
- defined, 6-3
- defining
  - for multiple partitions, 6-4
  - for replicated partitions, 16-6
  - for transparent partitions, 16-21, 16-32
- displaying, 16-41
- logging into, 16-7
- losing connections to, 16-62
- mapping information, 6-6

targets (*Continued*)

- mapping members to data sources
  - linked partitions, 16-35, 16-37
  - replicated partitions, 16-11, 16-17
  - specifying specific areas, 16-42
  - transparent partitions, 16-25, 16-27
- member names differing from source, 6-6
- missing data, 16-62
- partitioning information for, 6-6
- propagating outline changes, 16-6, 16-21, 16-32
  - process summarized, 16-47, 16-48
- selecting, 16-4
- specifying shared areas, 16-8, 16-22, 16-33
- updating changes to, 6-11, 16-5, 16-60
- viewing data in linked partitions, 6-23

.TCP files, 47-5

TCP/IP connections, 45-14

.TCT files, 40-16, 42-11, 47-3

.TCU files, 40-16, 47-3

technical support, 46-26

temporary files, 40-16

- deleting, 40-17

temporary values, 31-12, 32-18

terminating server connections

- caution, 17-33

- for specific user, 17-35, 17-38

terminology, 3-5

testing

- calculations, 33-2
- database design, 5-25
- for missing values, 27-5
- partitions, 6-7, 16-41

text

- See also* annotating; comments

- adding to calc scripts, 31-39
- adding to empty fields, 22-15
- attribute dimension type, 9-14
- attribute type, 10-6
- case conversions, 20-11, 22-16
- copying and pasting, 26-18, 31-41, 35-20
- cutting, 26-18, 31-41, 35-20

- text (*Continued*)
  - defining as labels, 37-26
  - displaying in Outline Editor, 8-26
  - fields, 22-10
  - finding and replacing
    - in formulas, 26-19
    - in report scripts, 35-17, 35-18
  - in data sources, 20-2
  - inserting in formulas, 26-15
  - linked reporting object, 12-1, 12-2
  - replacing missing values with, 36-25, 37-4
  - storing, 12-3
  - strings
    - replacing in rules file, 22-14
- TEXT command
  - usage, 36-18, 36-24
  - usage example, 37-25
- text editors
  - calc scripts and, 31-10
  - ESSCMD commands and, 44-10
  - formulas and, 26-11
  - report scripts and, 35-11
- text files, 47-3
  - calc scripts in, 31-24
  - creating, 39-1, 39-7
  - cross-platform compatibility, 47-16
  - data sources, 20-8
  - default locations, 21-6
  - dumping security information to, 45-5, 45-12
  - exporting outlines to, 8-11
  - loading, 23-2, 23-5
    - multiple, 23-7
  - opening, 21-5
- text masks, 36-23, 36-27
- text strings. *See* strings
- The Beverage Company (TBC), 5-3, 43-2
- third-party backup utility, 48-2, 48-3
- threshold (transactions), 42-10
- tildes (~)
  - as codes in data source, 14-25
  - in headings, 36-13
- time balance first/last properties
  - described, 5-30, 9-5
  - in combination with skip property, 9-7
  - setting, 9-5
- time balance properties
  - described, 5-29
  - examples for usage, 9-4, 9-5
- time balance tags, 30-1
  - calculating accounts data, 30-1
- time balanced data
  - calculating averages, 30-1
  - missing and zero values, 30-5
  - specifying in data sources, 14-25
- Time dimension
  - currency applications, 43-3
  - time series members and, 30-9, 30-14
  - usage example, 4-14
- time dimension
  - calculating values, 28-23, 30-1
  - defining formulas for, 30-6
  - description, 5-20
  - setting, 9-2
  - specifying, 9-2, 9-3, 30-2
  - time balance members and, 9-4
  - two-pass calculations and, 33-28
  - usage example, 5-28, 5-30
- time-out errors, 24-2, 24-5
- time-out settings
  - locks, 17-31
  - transactions, 41-16
- time periods, 5-20
  - budgeting expenses for, 9-9
  - determining first value for, 30-4, 30-6
  - determining last value for, 30-3, 30-6
  - getting average for, 30-4, 30-6
- time-sensitive data, 5-16
- time series reporting, 30-1
  - calculating averages, 30-4
  - calculating period-to-date values, 30-7
  - creating dynamic members for, 30-7, 36-32
  - getting first/last values, 30-3, 30-4, 30-6
  - partitioned databases, 30-16
  - retrieving period-to-date values, 30-14
  - skipping missing or zero values, 30-5
- Time Series tags
  - Intelligent Calculation and, 34-18
  - listed, 30-5
- time tags, 4-8
- time zones, 6-9

- time-out settings
  - locks, 42-7, 42-9
  - transactions, 42-2
- timestamps, comparing, 16-49
- TIMINGMESSAGES parameter, 46-30
- titles, 35-8, 36-24
- @TODATE function, 10-37, 26-46
- TODATE command, 36-36
  - usage, 36-28
- tokens, 22-2
- toolbar
  - Data Prep Editor, 21-4
  - Formula Editor, 26-15
  - Outline Editor, 8-4, 9-18
    - enabling/disabling, 8-25
  - Script Editor, 31-39
- tools, 1-2
- TOP command
  - caution for usage, 36-50
  - entering in report scripts, 36-46, 36-48, 36-51
  - precedence, 36-46
  - restrictions, 36-51
  - upper limits, 36-50
  - usage, 36-45
  - usage example, 37-53
- TOP.REP, 37-53
- top-down calculation, 33-12, 33-13
- top-down ordering
  - calculations, 9-17
  - dynamic builds, 13-3, 13-5, 13-6, 13-7
- top-down partitioning, 6-33
  - defined, 6-2
- totals, 36-16, 36-20
- trace files, 46-27
  - creating, 46-33, 46-35
- tracing calculations, 33-4
- traffic coordinator, 45-12
- trailing spaces, 22-18
- transaction control files, 47-3
- Transaction Control Table (TCT), 40-6, 42-1
- Transaction Manager, 40-2
  - overview, 40-6
- Transaction page (Database Settings), 41-6, 41-17
  - accessing locked blocks, 20-24
- transactions
  - actions triggering, 40-6
  - canceling calculations and, 25-11
  - caution for committed access, 41-16
  - caution for data loads, 24-5
  - committing, 40-6, 42-9
  - defined, 42-1
  - forcing at load completion, 22-23
  - initiating commits, 42-10
  - locks and, 42-6, 42-7, 42-9
  - managing, 40-6
  - multiple, 42-4
  - not completing, 42-12
  - predictability, 42-4
  - processing, 42-4
  - required privileges, 42-3
  - rolling back, 42-11, 45-3, 45-9
  - server crashes and active, 42-11
  - setting isolation levels
    - with Application Manager, 41-17
    - with ESSCMD, 41-7, 41-18
    - with Hyperion Essbase kernel, 41-15, 41-16
  - tracking, 40-6
  - updating isolation levels, 41-19
  - wait intervals, 41-16, 42-2
- transparent members, 38-8
- transparent partitions
  - advantages, 6-18
  - calculating, 6-21
  - clearing values in, 22-24
  - creating, 6-16, 16-20, 16-21, 16-22, 16-25
  - currency conversions and, 36-51, 43-26, 43-29
  - data loading and, 20-3
  - defined, 6-10
  - described, 6-15
  - disadvantages, 6-18
  - dynamically calculated values in, 29-26
  - example of, 6-35
  - formulas and, 6-21, 6-22
  - guidelines for selecting, 6-20, 6-27
  - implementing security measures, 6-31
  - improving performance, 6-20, 6-21, 6-25
  - mapping members, 16-25, 16-27
  - port usage, 6-23
  - usage restrictions, 6-17
  - using attributes in, 6-20

- .TRC files, 46-27, 46-33, 47-3
- tree icon, 8-22
- trees
  - See also* branches
  - data hierarchies, 3-6
  - moving members in, 14-10
  - outlines, 8-3
- @TREND function, 26-43
- trend analysis, 1-2
- trends, calculating, 26-43
- troubleshooting, 45-16, 46-26
  - calc scripts, 31-48
  - connections, 24-4
  - data loading, 24-1
  - formulas, 26-25
  - partitions, 16-62
- @TRUNCATE function, 26-39
- truncated names, 36-13
- truncating values, 26-39
- two-dimensional
  - data blocks, 4-14
- two-dimensional reports, 39-13, 39-14
- two-pass, 33-24
- two-pass calculation property, 9-15
  - usage example, 5-34
- two-pass calculations, 28-23, 33-24
  - as default, 33-27
    - examples, 33-29, 33-30
  - calc scripts and, 31-11, 33-31, 33-32, 33-33, 33-34
  - dynamic calculations and, 29-8, 29-17
  - Intelligent Calculation and, 33-32, 33-33
  - setting up, 9-15, 14-25
  - usage examples, 5-34
- .TXT files. *See* text files
- typefaces, 8-26
- types
  - See also* dimension types; field types
  - setting attribute, 9-14
  - tagging dimensions, 9-2
- typographical conventions, xlv

## U

- UCHARACTERS command, 36-22
- UCOLUMNS command, 36-22
- @UDA function, 26-37
- UDA command
  - selecting members with, 36-37
  - usage example, 36-33
- UDA field type
  - in header records, 14-24
  - in rules files, 14-15
  - nulls and, 13-7, 13-10
- UDAs
  - adding to report scripts, 36-37
  - allowing changes, 14-10
  - calc script example, 31-55
  - checking for, 26-30
  - compared with attributes, 10-9
  - creating, 9-26
  - described, 9-25
  - design approach for attributes, 10-13
  - flipping values in data load, 22-27
  - rules for creating, 9-25
  - shared members and, 9-23
- UDATA command, 36-22
  - usage example, 37-18
- UNAME command, 36-22
- UNAMEONDIMENSION command, 36-22
- unary operators, 5-26, 5-27
  - usage overview, 28-5, 28-7
- unauthorized users, 13-46, 20-24
- unavailable server or data sources, 24-1, 24-4
- uncommitted access
  - about, 42-3
  - commits and, 42-10
  - defined, 41-15
  - handling transactions with, 42-4
  - locks and, 40-6, 42-6
  - rollbacks and, 42-12
  - setting, 41-16, 41-17, 41-18
- UNDERLINECHAR command
  - usage, 36-22
  - usage example, 37-31

- underlining, 36-22
- UNDERSCORECHAR command, 37-31
- underscores ( \_ )
  - converting spaces to, 20-11, 22-19
  - in dimension and member names, 8-15
  - in report scripts, 36-3
- undoing
  - calc scripts builds, 31-36
  - database operations, 22-11
- unexpected values, 40-8
- Uniform Resource Locators. *See* URLs
- unique data values, 4-5
  - assigning #MISSING values to, 33-35
  - Intelligent Calculation and, 34-10
- UNIX platforms, 1-4
  - basic memory requirements, 15-14
  - file naming conventions, 47-16
  - freeing memory, 46-6
  - monitoring applications, 45-16, 46-41
  - running batch files, 44-9
  - specifying disk volumes, 41-26
  - starting Essbase server, 45-2
  - storing trace files, 46-34
- unknown member errors, 36-40
- unknown values, 20-15
- UNLOADALIAS command, 11-13
- UNLOADAPP command, 7-5, 45-7
- UNLOADDB command, 7-6, 45-9
  - usage overview, 45-8
- unlocking databases, 17-37
  - after allotted maximum lock time, 17-37
  - with Application Designer privilege, 17-35
- unlocking objects, 46-40, 47-15
- UNLOCKOBJECT command, 46-40
- unspecified members, 18-19
- Untitled report script, 35-12
- update log files, 48-8
- UPDATECALC setting, 34-5
  - system failures and, 48-7
- UPDATEFILE command, 23-1
- updates, 1-3, 42-9
  - getting information about outline, 46-17
  - troubleshooting, 16-63
- updating
  - alias tables, 14-21
  - cache, 41-9, 41-10, 41-12
  - Cache Memory Locking, 41-13
  - cache size, 41-9
  - changed blocks only, 34-1
  - data compression, 41-29
  - data sources, 6-15, 16-60
  - data targets, 6-11, 16-5, 16-60
  - databases, 13-46, 17-37
  - dimensions, 14-26
  - index page size, 41-15
  - indexes, 40-15
  - log files, 16-53
  - outlines, 23-12
  - partitioned applications, 6-7, 6-11, 6-13, 16-60
    - guidelines, 16-58
    - with remote data, 6-15
  - references, 18-16
  - reports, 35-11
  - requests. *See* transactions
  - spreadsheets, 48-8
  - substitution variables, 7-18
  - transaction isolation settings, 41-19
  - volumn settings, 41-26
- upgrades, 2-1, 41-2
- upper level blocks, 28-3, 28-25
  - dynamically calculating, 29-7, 29-14, 29-25
  - recalculating values in, 34-11
  - restructuring and, 40-28
- URLs
  - linking to cells, 12-1, 12-2
  - manipulating with Application Manager, 12-8
  - maximum character length, 12-5
  - storing, 12-3
- Use Aliases option, 26-25, 31-48
- Use Dimension Property Settings option, 14-21
- User Database Access dialog box, 17-25
- user groups
  - assigning filters to, 18-15
  - assigning privileges, 17-5
  - creating, 17-11, 17-14, 17-16

- user groups (*Continued*)
    - defined, 17-4, 17-14
    - defining security settings, 17-7, 17-9, 17-16
    - deleting, 17-20
    - editing, 17-14
    - modifying access settings, 17-12, 17-22, 17-27, 17-28
    - removing members, 17-11
    - renaming, 17-21
    - returning members, 17-11, 17-16
    - viewing current settings, 17-27, 17-28
  - user interface, 1-5, 7-7
    - See also* Application Manager
    - accessing linked objects and clients, 12-4
    - customizing, 1-5
  - user-management tasks, 17-7, 17-39, 17-41
  - user names
    - disabling, 17-10, 17-13, 17-18, 17-41
    - entering, 16-6, 17-10, 17-17
  - user types
    - Create/Delete Users/Groups, 17-6
    - defined, 17-5
    - ordinary, 17-6
    - Supervisor, 17-5
  - User/Group Application Access dialog box, 17-23
  - User/Group Security dialog box
    - copying security profiles, 17-17, 17-18
    - creating new users, 17-10, 17-14
    - deleting users and groups, 17-19, 17-20
    - editing existing groups, 17-14
    - editing user profiles, 17-12
    - managing users and groups, 17-8
    - renaming users and groups, 17-21
    - updating users, 17-12
  - user-defined attributes *See* UDAs
  - users, xli
    - accessing locked blocks, 20-24, 33-23
    - activating disabled, 17-41
    - assigning
      - application access, 17-11, 17-13, 17-22, 17-24
      - filters, 18-15
      - privileges, 17-5
      - to groups, 17-11, 17-15
  - users (*Continued*)
    - changing access privileges, 17-22, 17-27, 17-28
    - changing passwords, 17-13
    - creating accounts for, 6-26, 6-32
    - creating new, 17-9
    - defining security, 17-7, 17-9, 17-16
    - disconnecting from servers, 17-35, 17-38
    - displaying
      - access settings, 17-27, 17-28
      - defined users, 17-8
      - inactive users, 17-41
    - displaying current, 45-4, 45-10
    - editing security settings, 17-12
    - limiting login attempts, 17-39
    - logging out, 17-33, 17-39, 45-4, 45-10
    - login procedure, 44-7
    - maintaining information for, 7-4
    - removing, 17-11, 17-19
    - renaming, 17-21
    - replicating privileges, 17-16
    - security settings, 17-4
    - security types, 17-5
    - setting global access levels, 17-35
    - unauthorized, 13-46, 20-24
  - USERS command, 45-4
    - usage overview, 45-10
  - Users/Groups list box, 17-28
- ## V
- V, Dynamic Calc and Store code in data source, 14-25
  - VALIDATE command, 40-19, 42-13
    - partitioned applications and, 6-19
  - Validate page (Partition Wizard), 16-38
  - Validate Rules dialog box, 14-22, 21-23
  - VALIDATEPARTITIONDEFFILE command, 16-39
  - validating
    - data integrity, 42-13
    - field names, 21-23
    - partitions, 16-38
    - rules files, 14-21, 21-21, 21-23
    - troubleshooting problems, 24-6

- validity checking, 42-13
- values, 3-8, 25-1
  - See also* missing values; range of values
  - accumulation, 26-45
  - assigning to member combinations, 26-46
  - assigning to variables, 26-48
  - averaging, 9-6
    - for time periods, 30-1, 30-4
    - non-zero values and, 30-5
    - reporting examples, 37-32, 37-47
    - with formulas, 26-38, 27-3
  - changing, 22-22
  - comparing, 3-11, 9-9, 38-5
  - compression and repetitive, 40-8, 40-9, 40-10
  - defined, 3-9
  - displaying specific, 4-6
  - distribution among dimensions, 4-2
  - duplicating, 5-22
  - dynamically calculating, 29-2, 29-6, 29-15
  - entering in empty fields, 22-15
  - filtering, 18-2
  - flipping, 22-27
  - formatting in reports, 36-23, 36-26
    - examples, 37-17
  - identical, 5-28
  - in a database, 6-2, 35-8
  - inconsistent, 40-8
  - incorrect, 24-2
  - interdependent, 26-33
  - iterating through, 26-44
  - looking up, 26-4
  - measuring, 5-20
  - member with no, 9-22
  - negative, 20-4
    - flipping, 22-27
    - variance as, 26-39
  - nulls, 13-7, 13-10
  - optimizing in sparse dimensions, 24-8
  - ordering in reports, 36-45, 36-46
    - example, 37-55
  - out of range, 20-20
  - overwriting, 22-23, 23-17, 30-6
    - for currency conversions, 43-26
  - placing retrieval restrictions, 36-46, 36-50
    - example, 37-54
- values (*Continued*)
  - reading multiple ranges, 20-22
  - referencing, 3-8, 25-6, 26-49, 33-6, 33-7
  - retrieving dynamically calculated, 29-4, 29-10, 29-16, 29-21
  - retrieving for reports, 35-6
    - setting maximum rows allowed, 36-50
    - with conditions, 36-45, 36-48, 37-54, 37-55
  - retrieving from remote databases, 6-15, 29-14, 29-25
  - rolling up, 5-26
  - rounding, 24-12, 26-38
  - scaling, 22-25
  - setting maximum number of, 3-14
  - storing, 9-20, 9-23
  - temporary, 31-12, 32-18
  - truncating, 26-39
  - unable to change, 16-5
  - unexpected, 40-8
  - unique, 4-5
    - assigning #MISSING values to, 33-35
    - Intelligent Calculation and, 34-10
  - unknown, 20-15
  - variables as, 7-14
  - void, 16-43
  - @VAR function, 5-31, 9-9, 26-39
  - VAR command, 31-12
  - variables, 7-14
    - adding to report scripts, 36-33, 36-35, 36-36, 36-45
    - assigning values, 26-48
    - creating substitution, 7-15
    - declaring, 31-12, 31-52, 32-18
    - deleting, 7-17
    - free-form reports and, 35-30
    - inserting in calc scripts, 31-12, 31-52
    - setting configurable variables, 38-2
    - updating, 7-18
  - @VARIANCE function, 26-41
  - variance, 1-2, 5-18
    - See also* statistical variance, calculating
    - dynamically calculating, 29-17
    - returning, 26-39
    - usage examples, 5-3, 9-28, 26-8, 32-2

- variance percentages
  - calculating, 32-2
  - returning, 26-39
- variance reporting properties, 5-31, 30-1
  - setting, 9-9
- @VARIANCEP function, 26-41
- @VARPER function, 5-31, 9-9, 26-39
- Verify dialog box, 8-7
- verifying
  - See also* validating
  - outline content on save, 8-7
  - outlines, 8-5
  - path information, 46-5
  - values retrieved in loads, 24-3
- VERSION command, 45-5
  - usage overview, 45-12
- version compatibility, 8-8, 8-10
- version numbers, 45-5, 45-12
- vertical bars (|)
  - in application and database names, 7-12
  - in dimension and member names, 8-15
- View Log File dialog box, 29-11, 31-9, 46-28, 46-31
- View menu, 8-25, 21-4
- viewing
  - access settings, 17-27, 17-28
  - active filters, 18-3
  - active user groups, 17-27
  - aliases, 8-25, 11-6
    - in Calc Script Editor, 31-48
    - in Formula Editor, 26-25
  - application event log, 46-28
  - application information, 46-2, 46-8, 46-28, 46-31
  - applications, 21-2
  - available ports, 45-5, 45-10
  - changes to outlines, 46-35
  - current users, 17-8, 45-4, 45-10
  - data, 3-2, 3-8, 3-11
    - in targets, 6-23
  - with Data Prep Editor, 21-3, 21-4
  - data sources, 21-3, 21-4
  - data targets, 16-41
  - database information, 46-2, 46-9
  - database statistics, 33-2
  - databases, 21-2
- viewing (*Continued*)
  - dimensions
    - in Calc Script Editor, 31-47
    - in Formula Editor, 26-24
    - in outlines, 8-3
  - dynamically calculated members, 29-6, 29-11
  - field operations, 20-10
  - formulas, 8-25, 26-13
  - inactive users, 17-41
  - informational messages, 33-4, 33-23
  - linked objects, 12-6, 12-8
  - locked data, 42-7
  - locks, 17-37
  - log files, 29-10, 29-11, 31-9, 46-28, 46-31
  - member combinations, 4-4, 4-7
  - member names in reports, 36-43
  - members in outlines, 8-3, 29-6
  - names for generations and levels, 8-24
  - partitions, 16-57, 16-58
  - records, 21-20
  - replace operations, 20-11
  - selection/rejection criteria, 20-12
  - Server Agent commands, 45-5
  - server information, 46-3
  - software version, 45-5, 45-12
  - specific values, 4-6
  - storage information, 4-8
  - text in Outline Editor, 8-26
  - unique values, 4-5
- views, 3-2
  - customizing, 8-25
- virtual cubes (in Hyperion MBA). *See* transparent partitions
- virtual memory, 46-6
- Visual Basic programming objects, 47-6
- void mappings, 16-43
- volume names, 41-21
- volumes
  - allocation and, 40-4
  - data files and, 40-5
  - data storage and multiple, 40-11
  - deallocating, 41-22, 41-26
  - index files and, 40-3

volumes (*Continued*)

- specifying
  - with Application Manager, 41-22
  - with ESSCMD, 41-24
  - with Hyperion Essbase kernel, 41-20
- updating storage settings, 41-26

**W**

- .W95 files, 47-5
- Wait settings
  - locks, 42-7, 42-9
  - transactions, 41-16, 42-2
- warnings
  - partition validation, 16-39
  - report processing, 35-22, 35-25
- week-to-date calculations, 30-9
- white space
  - as file delimiter, 20-6
  - converting to underscores, 20-11, 22-19
  - cross-dimensional operator and, 26-46
  - data entry, 20-4
  - dropping, 20-11, 22-18
  - formulas, 26-6
  - in dimension and member names, 8-16
  - in names, 20-5
  - in report layouts, 36-26
  - report scripts, 36-2
- WIDTH command, 36-9
- wildcard matches, 31-55
- wildcards in report scripts, 36-38
- window (application), 7-7
- Windows Explorer, 23-7
- Windows File Manager, 23-7
- Windows platforms, 39-1
  - freeing memory, 46-6
  - monitoring applications, 45-16, 46-41
  - prerequisite user understanding, xlv
  - storing trace files, 46-34
- @WITHATTR function, 10-37, 26-38
- WITHATTR command, 6-28, 36-36, 37-60
  - usage, 36-28
- wizards, 16-3
- .WK4 files, 47-3
- workstations, 7-2
- write access, 13-46

- Write locks, 42-5
  - described, 42-5
  - with committed access, 42-2, 42-7
  - with uncommitted access, 42-3, 42-7
- Write privilege, 17-30, 18-2, 18-16
- writes, 1-3
  - data compression and, 40-8, 40-10
  - error log files, 46-27
  - getting number of, 46-14
  - optimizing, 24-11
- W-T-D time series member, 30-9, 30-14

**X**

- X member code, 29-23
- X operator, 26-16, 31-39
- X, Dynamic Calc code in data source, 14-25
- XCHGRATE database, 43-1, 43-3
  - adding members, 43-22
  - contents described, 43-5
  - displaying outlines, 43-21
  - linking, 43-24
- Xchgrate database
  - provided with Hyperion Essbase, xlv
- .XCP files, 46-27, 47-3
- .XLL files, 47-4
- .XLS files, 47-3
- @XREF function, 26-45

**Y**

- year-to-date calculations, 27-3, 30-8
- Y-T-D time series member, 30-8, 30-14

**Z**

- Z, time balance codes in data source, 14-25
- zero values, 33-35
  - excluding in time balances, 14-25
  - formatting in reports, 36-14, 36-23
  - including in time balances, 14-25
  - replacing with labels, 36-25
  - skipping, 9-6, 30-5
- ZEROTEXT command, 36-25
- zoom, 3-2