

Oracle® Hyperion Planning

Administrator's Guide



Release 11.2

F12985-03

May 2024

The Oracle logo, consisting of a solid red square with the word "ORACLE" in white, uppercase, sans-serif font centered within it.

ORACLE®

Oracle Hyperion Planning Administrator's Guide, Release 11.2

F12985-03

Copyright © 2001, 2024, Oracle and/or its affiliates.

Primary Author: EPM Information Development Team

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Documentation Accessibility

Documentation Feedback

1 Using Planning

About Planning	1-1
Planning Features	1-1
Important Information on New Features	1-2
Smart View	1-2
Planning Web Client	1-2
EPM Workspace	1-2
Lifecycle Management	1-2
Essbase	1-3
Financial Reporting	1-3
Business Rules	1-3
Predictive Planning	1-3
User Licensing for Third-Party Software	1-4

2 Getting Started with Planning

About Planning Applications	2-1
Starting Essbase	2-2
Considerations for Working with Essbase	2-2
Understanding Implied Sharing in Planning Forms	2-3
Connecting to Multiple Instances of Essbase	2-3
Starting the Relational Database	2-3
Optimizing SQL Relational Databases	2-4
Setting Application and System Properties	2-4
Changing the JDBC Driver	2-6
Configuring JDBC Connection Pooling	2-7
Controlling Smart View Messages	2-7
Showing New Planning Members in Smart View	2-8

Setting the Application Home Page	2-8
Setting Background Processing	2-8
Setting Business Rules Launch Logging Properties	2-10
Setting Data Validation Properties	2-11
Setting the Maximum Length for Text Values and Comments in Cells	2-11
Controlling Smart View Cell Formatting in Planning Forms	2-12
Setting Threshold Limits for Applications	2-12
Logging On to EPM Workspace and Accessing Planning	2-15
Working with Planning Utilities	2-16
Suppressing Password Prompts in Planning Utilities	2-17
About Text Files for Planning Utilities	2-18
Running Planning Utilities on UNIX	2-18
About EPM Oracle Instance	2-18
Using Application Servers	2-18
About Updating Instances and Clusters	2-19
Closing Planning and Essbase	2-19

3 Setting Up Access Permissions

Levels of Access Permissions	3-1
Planning Artifacts That Can Be Assigned Access	3-1
Types of Access Permissions	3-2
Enabling Access Permissions for Dimensions	3-3
Assigning Access to Members and Business Rules	3-3
Adding, Editing, and Removing Access	3-4
About Effective Access Permissions to Shared Members	3-5
Managing Access to Forms and Folders	3-6
Assigning Access to Forms and Folders	3-6
Adding, Changing, and Removing Access to Forms and Folders	3-7
Importing Access Permissions	3-8
Improving Performance When Importing Access Permissions	3-12
Exporting Access Permissions	3-13
Reporting on Access Permissions	3-16
Selecting Reporting Objects	3-16
Selecting Reporting Options	3-16
Working with Access Permissions Reports	3-17
Setting Up Audit Trails	3-17
Examples of Tracked Actions	3-19
Viewing and Clearing Audit Reports	3-19
Managing Security Filters	3-19
Synchronizing Users With the Provision Users Utility	3-20
Migrating User and Group Identities	3-21

Migrating Business Rule Security	3-22
Removing Stale User Records	3-23
Setting Up Access Permissions in Financial Reporting	3-24

4 Managing Planning Databases

Unlocking Applications	4-1
Using Broadcast Messaging	4-1
Viewing Usage Statistics	4-2
Creating and Refreshing Application Databases	4-2
Managing Exchange Rates	4-5
Creating Exchange Rate Tables	4-5
Editing Exchange Rate Tables	4-6
Deleting Exchange Rate Tables	4-7
Managing Currency Conversions	4-7
Working with Currency Conversion Calculation Scripts	4-8
Optimizing Performance	4-8
About Reordering Dimensions	4-9
Allocating Memory for Supporting Detail Cache	4-9
Configuring the Data and Index Cache Sizes	4-9
Other Performance Optimization Tips	4-10
Backing Up Applications and Application Databases	4-10

5 Importing and Exporting Data and Metadata

Overview	5-1
About Loading Dynamic Members	5-1
Loading Tools	5-1
Working with the Outline Load Utility	5-2
Command Properties File	5-3
Generating Load Files	5-4
Generating a Data Load File	5-4
Generating a Metadata Load File	5-6
Comments in Load Files	5-7
Load File Considerations	5-8
Importing Data and Metadata Using a Flat File	5-9
Loading Data	5-9
Loading Metadata	5-11
Importing Data and Metadata from a Relational Data Source	5-11
Importing Metadata	5-12
Importing Data	5-15
Exporting Data and Metadata from a Planning Application to a Flat File	5-19

Exporting Data with the Outline Load Utility	5-19
Exporting Metadata with the Outline Load Utility	5-20
Exporting Metadata from a Planning Application to a Relational Data Source	5-21
Command Line Parameters for the Outline Load Utility	5-24
Dimension Properties	5-36
Common Member Properties	5-37
Account Dimension Properties	5-39
Entity Dimension Properties	5-43
Period Dimension Properties	5-44
User-Defined Dimension Properties	5-45
Year Dimension Properties	5-45
Scenario Dimension Properties	5-46
Version Dimension Properties	5-46
Currency Dimension Properties	5-47
Attribute Dimension Properties	5-48
UDA Dimension Properties	5-48
Exchange Rate Dimension Properties	5-49
Planning Unit Hierarchy Dimension Properties	5-49
Smart List Dimension Properties	5-58
Working with Planning Import and Export	5-59
Running Planning Import and Export	5-59
Importing Metadata from a File	5-60
Importing Data from a File	5-62
Exporting Metadata to a File	5-64
Exporting Data to a File	5-64
Clearing EPM Oracle Instance Tmp Directory Files	5-65
Enabling Data Load of Essbase Data	5-65
Loading from Administration Services	5-66
File Example	5-67

6 Managing Forms

About Forms	6-1
Form Components	6-1
Point of View	6-1
Page Axis	6-1
Rows and Columns	6-1
Form Design Considerations	6-2
Forms and Plan Types	6-2
Forms and Access Permissions	6-2
Forms and Currencies	6-2
Forms and Versions	6-2

Forms and Attributes	6-2
Forms and Shared Members	6-2
Forms and Calculations	6-3
Creating Simple Forms	6-3
Setting Form Layout	6-4
Setting Form Grid Properties	6-5
Setting Dimension Properties	6-7
Setting Display Properties	6-8
Setting Printing Options	6-9
Including Data Validation Rules in Forms	6-10
Setting Form Precision and Other Options	6-11
Creating Asymmetric Rows and Columns	6-13
Adding Formula Rows and Columns	6-13
Defining Simple Form Page and Point of View	6-14
Importing Form Definitions	6-15
Preparing Forms	6-15
Preparing Data Files	6-15
Importing Form Definitions	6-16
ImportFormDefinition Utility Parameters	6-17
ImportFormDefinition Examples	6-17
Creating Composite Forms	6-18
Setting Composite Form Layout	6-18
Adding Simple Forms to a Composite Form Layout	6-20
Rearranging Forms in the Layout	6-20
Editing Simple Forms from within a Composite Form	6-20
Deleting Simple Forms from a Composite Form	6-21
Setting Composite Form Section Properties	6-21
Setting Composite Form Point of View and Page Dimensions	6-22
Creating Master Composite Forms	6-23
Embedding Charts in Composite Forms	6-24
Designing Specific Types of Forms	6-27
Designing Forms for Multiple Currencies	6-27
Designing Forms for Drill-Through Information	6-27
Designing Forms with Formula Rows and Columns	6-27
Designing Forms with Data Validation	6-28
Designing Forms with Global Assumptions	6-28
Designing Forms for Rolling Forecasts	6-28
About Rolling Forecasts	6-28
Creating Rolling Forecasts	6-29
Modifying Rolling Forecast Variables	6-31
Working with Forms and Form Components	6-31
Selecting and Opening Forms and Folders	6-31

Previewing Forms	6-32
Printing Form Definitions	6-32
Searching for Forms	6-33
Editing Forms	6-33
Editing Simple Forms	6-33
Editing Composite Forms	6-34
Moving, Deleting, and Renaming Forms	6-34
Importing and Exporting Form Definitions	6-35
Specifying How Many Forms Users Can Simultaneously Use	6-36
Using Grid Diagnostics	6-37
Using Business Rules	6-38
About Business Rules	6-38
Selecting Business Rules	6-38
Setting Business Rule Properties	6-39
About Runtime Prompts	6-40
Understanding Runtime Prompts	6-40
About Runtime Prompts and Approvals Security	6-44
Designing Secure Runtime Prompts	6-45
Launching Business Rules With a Utility	6-46
Customizing Error Messages	6-48
How Cell Formatting in Smart View Persists in Planning	6-48
Working with Substitution Variables	6-49
About Substitution Variables	6-49
Creating and Assigning Values to Substitution Variables Using Planning	6-49
Deleting Substitution Variables Using Planning	6-50
Working with User Variables	6-50
About User Variables	6-50
Managing User Variables	6-50
Creating User Variables	6-51
Deleting User Variables	6-51

7 Using the Member Selector

About Working with Members	7-1
Selecting Members	7-1
Using Wildcards in Searches	7-3
About Selecting Attribute Values as Members	7-4
About Selecting Members for Forms	7-6
About Selecting Substitution Variables as Members	7-7
About Selecting User Variables as Members	7-9

8 Managing Data Validation

Creating and Updating Data Validation Rules	8-1
Formatting Cells and Setting the Promotional Path	8-3
Viewing Data Validation Rules	8-4
Order of Evaluation and Execution for Data Validation Rules	8-5
Conditions Supported by the Rule Builder	8-5
If Condition Values	8-6
Current Cell Value	8-6
Cell Value	8-7
Column Value	8-8
Row Value	8-9
Cross Dim Member	8-10
Member Name	8-11
Member	8-11
Account Type	8-12
Version Type	8-12
Variance Reporting Type	8-12
UDA	8-13
Attribute	8-13
Then Condition Values	8-14
Range Condition Values	8-14
Check Range	8-15
Range	8-15
Data Validation Conditional Operators	8-15
Data Validation Rule Scenarios	8-17
Scenario 1	8-17
Scenario 2	8-19
Scenario 3	8-19
Scenario 4	8-21
Scenario 5	8-21

9 Managing the Budgeting Process

About the Budgeting Process	9-1
Planning Units	9-1
Review Process	9-1
Planning Unit Hierarchy	9-2
Data Validation Rules	9-2
Task Lists	9-2
Setting Up Email for Approvals Notification	9-2
Defining the Budgeting Process	9-3

Approvals Roles	9-3
Approvals Process	9-3
Approval Operations and Data Validations	9-4
Starting and Supporting the Review Process	9-4
Printing Planning Unit Annotations	9-5
Managing Planning Unit Hierarchies	9-5
Creating Planning Unit Hierarchies	9-6
Setting Planning Unit Hierarchy Name, Scope, and Template	9-6
Selecting Planning Unit Hierarchy Members	9-7
Assigning Planning Unit Owners and Reviewers	9-9
About Group-based Approvals	9-10
Assigning Planning Unit Hierarchy Scenario and Version Combinations	9-11
Selecting Planning Unit Hierarchy Scenario and Version	9-12
Editing Planning Unit Hierarchies	9-12
Deleting and Renaming Planning Unit Hierarchies	9-13
Viewing Planning Unit Hierarchy Usage	9-13
Synchronizing Planning Unit Hierarchies	9-14
Exporting Planning Unit Hierarchies	9-15
Importing Planning Unit Hierarchies	9-16
Planning Unit Promotional Path	9-16
Modifying the Planning Unit Promotional Path	9-17
Design Considerations for Planning Unit Promotional Path Data Validation Rules	9-19
Managing Task Lists	9-19
Working with Task List Folders	9-19
Working with Task Lists	9-20
Adding Instructions to Task Lists	9-20
Adding and Defining Tasks	9-20
Adding Tasks to Task Lists	9-21
Editing Task Lists	9-23
Editing Tasks	9-23
Copying and Moving Tasks	9-24
Moving and Reordering Task Lists	9-24
Clearing Task Lists	9-24
Deleting Tasks and Task Lists	9-25
Linking Tasks to Planning Pages	9-25
Assigning Access to Task Lists	9-26
Adding Access to Task Lists	9-26
Changing and Removing Access to Task Lists	9-26
Importing and Exporting Task Lists	9-27
Copying Data	9-28
Clearing Cell Details	9-29

10 Working with an Application

Managing Forms and Folders	10-1
Creating Folders	10-1
Working with Folders	10-2
About Setting Preferences	10-2
Setting Personal Preferences	10-3
Setting Application Defaults	10-3
Specifying System Settings	10-4
Assigning Applications to Shared Services Application Groups	10-5
Limiting Use of an Application During Maintenance	10-6
Using the MaintenanceMode Utility to Limit Application Access	10-6
Specifying Custom Tools	10-8
Setting Display Options	10-8
Setting Printing Options	10-8
Mapping an Application for Reporting	10-8
About the Reporting Application	10-9
Creating a Reporting Application	10-9
Mapping a Planning Application to a Reporting Application	10-10
Defining Application Mappings	10-10
Defining Dimension Mappings	10-10
Setting the Point of View	10-11
Setting Data Options	10-12
Validation Rules for Default Members	10-12
Pushing Data	10-13
Pushing Data to a Reporting Application	10-13
Pushing Data with a Utility	10-14
Unsupported Features with Mapping Applications	10-15
Mapping Applications and Substitution Variables	10-15
Adding an Essbase Server for Reporting Applications	10-16
Editing an Essbase Server for Reporting Applications	10-16
Removing an Essbase Server for Reporting Applications	10-17
Deleting Application Information Using SQL	10-17
Deleting Account Annotations	10-18
Deleting Supporting Detail Associated With a Scenario	10-18

11 Working with Menus

Creating and Updating Menus	11-1
Working with Menu Items	11-1
Adding or Changing Menu Items	11-2

12 Working with Planning Application Administration

About Creating Applications with Planning Application Administration	12-1
Setting Up Applications Using Planning Application Administration	12-1
Managing Applications	12-1
Creating Applications	12-1
Selecting Application Information	12-2
Setting up the Calendar	12-2
Setting up Currencies	12-4
Specifying Plan Types	12-4
Reviewing Application Information	12-5
Deleting Applications	12-5
Registering Applications	12-6
Managing Data Sources	12-6
Creating Data Sources	12-6
Editing Data Sources	12-8
Checking Connections	12-8
Deleting Data Sources	12-9
Managing Upgrades	12-10
Updating References to Data Sources	12-10
Upgrading Applications	12-11
Updating References to Reporting Essbase Servers	12-11
About Monitoring and Optimizing Performance Using the Application Monitor	12-12
Assumptions	12-13
How the Application Monitor Works	12-13
Using the Application Monitor Graphs	12-14
Setting Limits for Application Monitor Warning and Error Thresholds	12-15
Launching the Application Monitor	12-15
Modifying Artifacts for Optimal Performance	12-16
Working with Alias Tables	12-17
About Alias Tables	12-17
Creating Alias Tables	12-17
Editing or Renaming Alias Tables	12-17
Deleting Alias Tables	12-18
Clearing Alias Tables	12-18
Copying Alias Tables	12-18
Specifying a Default Alias Table, and Setting Member and Alias Display Options	12-18
Working with Dimensions	12-19
Dimension Overview	12-19
About Dimensions and Members	12-19
About Sparse and Dense Dimensions	12-19
About Dimension Hierarchies	12-19

Working with Dimension Hierarchies	12-20
Expanding and Collapsing Dimension Hierarchies	12-20
Navigating Dimension Hierarchies	12-21
Filtering the Dimension View by Plan Type	12-21
Finding Dimensions or Members	12-21
Sorting Members	12-21
Moving Members Within the Dimension Hierarchy	12-22
Viewing a Member's Ancestors	12-22
Determining Where Members Are Used in an Application	12-22
About Custom Dimensions	12-23
Aggregation Options	12-23
Storage Options	12-23
About Entities	12-25
Base Currency	12-25
About Accounts	12-25
Account Types	12-25
Saved Assumptions	12-28
Data Type and Exchange Rate Type	12-29
Accounts, Entities, and Plan Types	12-29
Entities and Plan Types	12-29
Accounts and Plan Types	12-29
About User-Defined Custom Dimensions	12-29
Adding or Editing User-Defined Custom Dimensions	12-30
Setting Dimension Properties	12-30
Setting Dimension Density and Order	12-31
Setting the Evaluation Order	12-31
Working with Members	12-32
About Dynamic Members	12-32
Adding or Editing Members	12-33
Deleting Members	12-38
Deleting Parent Members	12-39
Viewing Member Properties from Forms	12-39
Working with Shared Members	12-39
Creating Shared Members	12-40
Working with Attributes	12-40
Understanding Attribute Data Types	12-41
Deleting Attributes	12-41
Working with Attribute Values	12-42
Creating Attribute Values	12-42
Assigning Attribute Values to Members	12-42
Modifying Attribute Values	12-43
Deleting Attribute Values	12-43

Customizing Calendars	12-43
Defining How Calendars Roll Up	12-44
Creating and Editing Summary Time Periods	12-44
Deleting Summary Time Periods	12-45
Working with the Years Dimension	12-45
Adding Years to the Calendar	12-45
Setting the Fiscal Year	12-46
Editing Year Information	12-46
Renaming Time Periods	12-46
Assigning Aliases to Summary Time Periods	12-47
Editing the BegBalance Member	12-47
Setting Up Currencies	12-47
Enabling Multiple Currencies	12-48
Working with Multiple Currencies	12-48
About the Calculate Currencies Business Rule	12-49
Exchange Rate Types	12-49
Scaling	12-49
Number Formatting	12-49
Reporting Currencies	12-50
Checking How Currencies are Used	12-50
Creating Currencies	12-50
Editing Currencies	12-51
Deleting Currencies	12-52
Specifying Exchange Rates	12-52
About Exchange Rate Tables	12-53
Hsp_Rates Dimension	12-53
Triangulation	12-53
Calculation Method	12-53
Setting Up Scenarios	12-54
About Scenarios	12-54
Creating Scenarios	12-55
Editing Scenarios	12-55
Deleting Scenarios	12-56
Copying Scenarios	12-56
Specifying Versions	12-56
About Versions	12-57
Target and Bottom Up Versions	12-57
Creating Versions	12-57
Editing a Version	12-58
Deleting Versions	12-58
Displaying Versions	12-59
Sorting Version and Scenario Members	12-59

Moving Scenario and Version Members in the Dimension Hierarchy	12-59
Adding an Aggregate Storage Outline to a Planning Application	12-59
About Aggregate Storage	12-60
Aggregate Storage Outline Plan Type Characteristics	12-60
Process for Adding an Aggregate Storage Database to a Planning Application	12-60
Adding Plan Types	12-61
Setting up Dynamic Time Series Members	12-61
Additional Supported Planning Application Features	12-63
Considerations for Alternate Hierarchies in Period Dimensions	12-64
Working with Applications that Use Planning Application Administration	12-64
Refreshing Application Databases Using a Utility	12-64
Refreshing Databases with Level 0 Members Set to Dynamic Calc	12-65
Working With Essbase Partitions	12-65
Sorting Members Using a Utility	12-66
Deleting Shared Descendants Using a Utility	12-67
Using Smart Lists, UDAs, and Member Formulas	12-68
Working with Smart Lists	12-68
Synchronizing Smart Lists in Reporting Applications	12-68
Adding or Changing Smart List Properties	12-69
Adding or Changing Smart List Entries	12-70
Previewing Smart Lists	12-71
Displaying #MISSING with Smart Lists	12-71
Working with UDAs	12-72
Creating UDAs	12-73
Changing UDAs	12-73
Deleting UDAs	12-73
Working with Member Formulas	12-73
Viewing Details of Formula Validation	12-74
Working with Formula Expressions	12-74

13 Customizing Planning Web Client

Customizing Reports	13-1
Customizing the Planning Tools Page	13-3
Customizing Cascading Style Sheets	13-3
Location of Cascading Style Sheet Files	13-4
Modifying Cascading Style Sheets	13-4
Customization Examples	13-5
Changing the Color of the Content Area Background	13-5
Changing the Color of Hyperlinks and Visited Hyperlinks	13-5
Customizing the Style of Row and Column Members in Forms	13-5
Customizing Skins to Add Components	13-6

Registering Customized Skins	13-6
About Customizing ADF Components	13-7
About Customizing Text, Color, and Images	13-7
Customizing Text, Color, and Images	13-7
Customizing Text in Non-Latin Languages	13-8
Restricted Characters in Customized Messages	13-10
Customizing Colors, States, and Actions for Process Status	13-10
Customizing Colors	13-10
Customizing States	13-11
Customizing Actions	13-11
Creating Spreading Patterns	13-12
Examples of Spreading Patterns	13-13
Creating Instructions for Planners	13-14

14 Troubleshooting

Calculation Script is Too Long or Complex	14-1
Cannot Find Member	14-1
Cannot Process Request	14-1
Conflicting Change by Users	14-2
Copy Version Error	14-2
Currency Conversion Calculation Script Failed	14-3
Form Error Message	14-3
Database Locked by Another Administrator	14-3
500 Error Message	14-3
Implied Sharing Issues	14-4
Manage Database Errors	14-5
Maximum Number of Applications Already Connected or Database is Locked	14-6
Optimizing WebLogic Server Parameters	14-7
Optimizing Windows Network Parameters	14-8
Registering a Restored Application with Planning	14-8
Session Timeout	14-9
Slow Performance When Opening Forms Using a Dial-Up Connection	14-9
Unable to Create Application Error	14-10
Unable to Refresh Application Error	14-11
Unsuccessful Log On	14-11

A Naming Restrictions

Restrictions for Applications and Databases	A-1
Restrictions for Dimensions, Members, Aliases, and Forms	A-1

Dimension and Member Names in Calculation Scripts, Report Scripts, Formulas, Filters, and Substitution Variables	A-4
Restrictions for User Names	A-5

B Form Formula Functions

Using Formulas and Formula Functions	B-1
Creating Formulas	B-1
Editing Formulas	B-1
Deleting Formulas	B-1
Formula Functions	B-2
Arguments	B-3
Numeric Arguments	B-4
Row, Column, or Cell Reference Arguments	B-4
Property Arguments	B-6
Embedded Functions as Arguments	B-8
Abs	B-8
Average	B-9
AverageA	B-10
Count	B-11
CountA	B-12
Difference	B-12
Eval	B-13
IfThen, If	B-14
Notes on Conditions	B-29
Complex Conditions	B-30
Max	B-30
Min	B-31
Mod	B-31
PercentOfTotal	B-32
Pi	B-33
Product	B-33
Random	B-34
Rank	B-34
Round	B-36
Sqrt	B-37
Sum	B-38
Truncate/Trunc	B-39
Variance/Var	B-40
Expected Results	B-40
Variance Behavior	B-40
Examples	B-41

VariancePercent/VarPer	B-41
Expected Results	B-42
VariancePercent Behavior	B-42
Examples:	B-42

C Custom-defined Functions

@HspNumToString	C-1
@HspDateDiff	C-1
@HspDateRoll	C-2
@HspDateToString	C-2
@HspGetMbrCount	C-3
@HspGetMbrIndex	C-3
@HspNthMbr	C-4
@HspStringCompare	C-4
@HspOffsetIdx	C-5
@HspStringCompareArray	C-5
@HspConcat	C-6

D Using Smart View to Import and Edit Planning Metadata

About Using Smart View to Import and Edit Planning Metadata	D-1
Importing Planning Dimensions in Smart View	D-1
Importing Dimensions in Smart View	D-2
Using Smart View Grids	D-2
Overview of the Smart View Grid	D-2
The Smart View Grid Display	D-2
Guidelines for Using the Smart View Grid	D-3
Editing Members in Smart View	D-3
Adding Planning Members in Smart View	D-4
Selecting the Add Mode within Planning Application Properties	D-4
Adding Members in Smart View	D-5
Guidelines for Adding Members in Smart View	D-6
Moving Members in Smart View	D-6
Guidelines for Moving Members in Smart View	D-7
Designating Shared Members in Smart View	D-7
Refreshing and Creating Cubes in Smart View	D-8

Documentation Accessibility

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

Access to Oracle Support

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

Documentation Feedback

To provide feedback on this documentation, click the feedback button at the bottom of the page in any Oracle Help Center topic. You can also send email to epmdoc_ww@oracle.com.

1

Using Planning

Related Topics

- [About Planning](#)
- [Planning Features](#)
- [Important Information on New Features](#)
- [Smart View](#)
- [Planning Web Client](#)
- [EPM Workspace](#)
- [Lifecycle Management](#)
- [Essbase](#)
- [Financial Reporting](#)
- [Business Rules](#)
- [Predictive Planning](#)
- [User Licensing for Third-Party Software](#)

About Planning

Use Oracle Hyperion Planning to drive collaborative, event-based planning processes. Decision-makers and managers collaborate with budget holders to optimize the planning process, adapting rapidly and ensuring optimal plans.

Planning Features

Oracle Hyperion Planning:

- Facilitates collaboration, communication, and control across multidivisional global enterprises
- Provides a framework for perpetual planning, to manage volatility and frequent planning cycles
- Provides ease of use and deployment through the Web or Oracle Smart View for Office
- Decreases the total cost of ownership through a shorter roll out and implementation phase, and easier maintenance for an application
- Enhances decision-making with reporting, analysis, and planning
- Promotes modeling with complex business rules and allocations
- Integrates with other systems to load data

Important Information on New Features

For important information on using Oracle Hyperion Planning new features, see *Oracle Hyperion Planning New Features* for the current release.



Note:

Planning no longer supports the Classic user interface (that is, up to and including Planning Release 11.1.2.1).

Smart View

Oracle Smart View for Office is the Microsoft Office interface to Oracle Hyperion Planning, where users can perform their planning and forecasting activities. They can analyze Planning data in Microsoft Outlook, Excel, Word, and PowerPoint.

Ad hoc grids in Smart View enable Planning users to personalize focused data slices that they frequently access and share them with others through Smart View or Planning. See the "Working with Ad Hoc Grids" chapter in the *Oracle Hyperion Planning User's Guide* and the *Oracle Smart View for Office User's Guide*.

Planning Web Client

Oracle Hyperion Planning provides complete functionality for Web users. Use the Web interface to roll out applications to large, distributed organizations without installing software on client computers. All software resides on the server. Many administrative features that were formerly in the Planning Desktop are now available through Planning Web.

EPM Workspace

Oracle Hyperion Planning is available within Oracle Hyperion Enterprise Performance Management Workspace. For information on EPM Workspace features, such as EPM Workspace preferences, see the *Oracle Hyperion Enterprise Performance Management Workspace User's Guide* or *Oracle Hyperion Enterprise Performance Management Workspace Administrator's Guide*. To log on to EPM Workspace, see [Logging On to EPM Workspace and Accessing Planning](#).

Lifecycle Management

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

Lifecycle Management features:

- Viewing applications and folders

- Searching for artifacts
- Migrating directly from one application to another
- Migrating to and from the file system
- Saving and loading migration definition files
- Viewing selected artifacts
- Auditing migrations
- Viewing the status of migrations
- Importing and exporting individual artifacts for quick changes on the file system

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

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

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

Essbase

Oracle Hyperion Planning leverages Oracle Essbase analytic and calculation capabilities, security filters, APIs, pre-built financial intelligence, calculation functions, and multi-database application support. Planning stores the application definition in a relational database, and creates Essbase databases and security privileges for applications.

Data sources are used to link the relational database and the Essbase server, and are associated with each Planning application. For information on creating data sources for Planning application administration, see [Managing Data Sources](#).

Financial Reporting

Oracle Hyperion Financial Reporting is a management reporting tool that transforms data into meaningful business information through highly formatted reports. Oracle Hyperion Planning users can use Financial Reporting to manage reporting tasks and implement plan review and analysis. Users can create reports featuring text, grids of data, charts, graphs, and images. They can use real-time, ad hoc variance reporting, and produce a variety of sophisticated financial reports that can be viewed online or printed with production-quality formatting.

Business Rules

Business rules perform complicated calculations through predefined formulas, variables, calculation scripts, macros, rulesets, and templates. See [Using Business Rules](#).

Predictive Planning

With Predictive Planning, users can work in Oracle Hyperion Planning forms to predict performance based on historical data. Predictive Planning uses sophisticated time-series and autoregressive integrated moving average (ARIMA) statistical techniques to confirm and

validate predictions entered into Planning based on other prediction methods. To use this feature, administrators must design forms as described in the *Oracle Hyperion Planning Predictive Planning User's Guide*. Then, if a valid Planning form is loaded in Oracle Smart View for Office, users can access Predictive Planning functionality from the Planning ribbon. (For installation instructions, see the *Oracle Enterprise Performance Management System Installation and Configuration Guide*.)

User Licensing for Third-Party Software

To use Oracle Hyperion Planning, you must purchase licenses from third-party vendors, for example, for a relational database and Web application server.

2

Getting Started with Planning

Related Topics

- [About Planning Applications](#)
- [Starting Essbase](#)
- [Considerations for Working with Essbase](#)
- [Understanding Implied Sharing in Planning Forms](#)
- [Connecting to Multiple Instances of Essbase](#)
- [Starting the Relational Database](#)
- [Optimizing SQL Relational Databases](#)
- [Setting Application and System Properties](#)
- [Logging On to EPM Workspace and Accessing Planning](#)
- [Working with Planning Utilities](#)
- [Using Application Servers](#)
- [About Updating Instances and Clusters](#)
- [Closing Planning and Essbase](#)

About Planning Applications

A Oracle Hyperion Planning application is a related set of dimensions and dimension members used to meet a set of planning needs. Each application has its own accounts, entities, scenarios, and other data elements.

This guide provides information about performing Planning application administration tasks, including:

- Creating, deleting, and registering Planning applications.
- Creating forms, task lists, and menus.
- Assigning access permissions for dimension members, task lists, forms, and form folders. See [Setting Up Access Permissions](#) and the *Oracle Enterprise Performance Management System User Security Administration Guide* and its information on Planning.
- Managing currency conversion and exchange rate tables.
- Identifying the review and approval process, requirements, and participants, and managing the budgeting process.

For information on installing and configuring Planning, see the *Oracle Enterprise Performance Management System Installation and Configuration Guide*.

Starting Essbase

The data for Oracle Hyperion Planning applications resides on the Oracle Essbase server. Essbase must be running before you can open Planning applications. After Essbase is started, it can be minimized and run in the background or as a service.

Considerations for Working with Essbase

Creating and refreshing on the Manage Database page affects data in the Oracle Essbase database. When you click Create, data is erased and Oracle Hyperion Planning plan types are rebuilt. When you click Refresh, data might be replaced. For important information, see [Creating and Refreshing Application Databases](#).

When you refresh Planning applications, the Essbase database makes these changes:

- If a member does not exist in the Planning relational database, the member and its properties, attributes, and User-Defined Attributes (UDAs) are not retained in Essbase on Refresh.
- Planning first retrieves member properties from any existing member in Essbase, then sets and overrides any member property definable in Planning.

This behavior is different from prior releases, where the refresh process was incremental and retained modifications made directly to the outline in Essbase, even where those members, attributes, and UDAs did not exist in Planning.

The `HSP_UDF` UDA preserves member formulas defined outside of Planning. You can assign this UDA to members from within Planning. Unassigning them through Planning refresh does not unassign the UDAs from members in the Essbase outline. If a member with the `HSP_UDF` UDA is added directly in Essbase, refreshing the database preserves any formula on this member in Essbase, and does not generate a formula placeholder (;) on members that are level zero and dynamic calculation. Other UDAs, such as custom UDAs and the `HSP_NOLINK` UDA, are retained only if defined in Planning application administration. To create and update UDAs, see [Working with UDAs](#).

Because member formula support is available in Planning application administration, there is less need for the `HSP_UDF` UDA. Every effort should be made to define UDAs in Planning application administration.

The `@XREF` function looks up a data value from another database to calculate a value from the current database. You can add the `HSP_NOLINK` UDA to members to prevent the `@XREF` function from being created on all plan types that are not the source plan type selected for that member. For more information on UDAs and functions, see the *Oracle Essbase Database Administrator's Guide*.

It is not necessary to modify Essbase outlines for the most commonly used Essbase member properties. Planning supports Dynamic Time Series in the Period dimension, alternate hierarchy in the Period dimension, hierarchies in Scenario and Version dimensions, and hierarchy and aliases for attribute dimensions.

Use these guidelines for working in Essbase while logged on as a supervisor or application administrator:

- Do not change dense/sparse dimension settings or the order of dimensions in Essbase.
- You can use any attribute dimension type, including text, Boolean, numeric, and date.
- Do not use Link Reporting Objects (LRO).

- Do not change the Time Balance default settings. The defaults are Ending for Balance and None for Flow.
- Do not change the Variance Reporting settings. These are preset for all account types except Saved Assumptions. For Saved Assumptions account types, Expense and Non-Expense are valid settings.
- Back up the database before refreshing. See [Backing Up Applications and Application Databases](#).

Understanding Implied Sharing in Planning Forms

With Oracle Essbase implied sharing, some members are shared even if you do not explicitly set them as shared. These members are implied shared members.

When an implied share relationship is created, each implied member assumes the other member's value. Oracle Hyperion Planning assumes (or implies) a shared member relationship in these situations:

- A parent has only one child
- A parent has only one child that consolidates to the parent
- The data type of the parent is Label Only (in this case, the parent inherits the value of the first child, regardless of the aggregation settings used for the child)

In a Planning form that contains members with an implied sharing relationship, when a value is added for the parent, the child assumes the same value after the form is saved. Likewise, if a value is added for the child, the parent usually assumes the same value after a form is saved.

For example, when a calculation script or load rule populates an implied share member, the other implied share member assumes the value of the member populated by the calculation script or load rule. The last value calculated or imported takes precedence. The result is the same whether you refer to the parent or the child as a variable in a calculation script.

When necessary, you can avoid implied sharing on Planning forms. See [Implied Sharing Issues](#).

Connecting to Multiple Instances of Essbase

You can connect to multiple instances of Oracle Essbase simultaneously from the Oracle Hyperion Planning application by specifying the port number:

- When creating a Planning application with the Planning Application Wizard (repeat for every data source name).
- As the value for the property `ESS_SERVER` in the `HSPSYS_DATASOURCE` database table.

For instructions on installing multiple instances of Essbase on the same computer, see the *Oracle Essbase Database Administrator's Guide*.

Starting the Relational Database

A relational database must be running before you can open Oracle Hyperion Planning applications. After the relational database is started, it can be minimized and run in the background or as a service.

Optimizing SQL Relational Databases

You can configure the timeout value through the Timeout registry key setting. Depending on the size of your database records, some SQL queries issued by Oracle Hyperion Planning could take longer to execute than the default ADO timeout of 30 seconds. This could lead to failure when refreshing the application database. You can increase the timeout value (for example, to 180 seconds) to decrease the likelihood of refresh failure.

To optimize the timeout value:

1. Open the Registry Editor.
2. Navigate to this location:

```
HKEY_LOCAL_MACHINE/SOFTWARE/HyperionSolutions/Planning
```

If the Timeout registry key exists in the right pane, the timeout value is set to a corresponding value. If the key does not exist, the value is set to the default. Its type must be DWORD.

Setting Application and System Properties

You can add or change application or system properties to customize such aspects as the path to the external authentication configuration file.

Table 2-1 Examples of Configurable Properties

PROPERTY_NAME Column	Description
RDB_DRIVER	See Changing the JDBC Driver .
RDB_SERVER_URL	
JDBC_MIN_CONNECTIONS	See Configuring JDBC Connection Pooling .
JDCB_MAX_CONNECTIONS	
SUPPORTING_DETAIL_CACHE_SIZE	See Allocating Memory for Supporting Detail Cache .
SUBST_VAR_CACHE_LIFETIME	See About Selecting Substitution Variables as Members .
DIRECT_DATA_LOAD	See Loading Data .
DATA_LOAD_FILE_PATH	
OLAP_MAX_CONNECTIONS	See Optimizing Performance .
SMART_VIEW_DISPLAY_WARNING	See Controlling Smart View Messages .
SMART_VIEW_MD_NEW_MEMBER_SUFFIX	See Selecting the Add Mode within Planning Application Properties .
SMART_VIEW_MERGE_FORMATTING	See Controlling Smart View Cell Formatting in Planning Forms .
HOME_PAGE	See Setting the Application Home Page .
RULE_MAX_WAIT, RULE_MONITOR_DELAY, CLR_CELL_MAX_WAIT, CLR_CELL_MONITOR_DELAY, COPY_DATA_MAX_WAIT, COPY_DATA_MONITOR_DELAY, PUSH_DATA_MONITOR_DELAY, PUSH_DATA_MAX_WAIT JOB_STATUS_MAX_AGE	See Setting Background Processing .

Table 2-1 (Cont.) Examples of Configurable Properties

PROPERTY_NAME Column	Description
BUSINESS_RULE_LAUNCH_LOG_ENABLED BUSINESS_RULE_LAUNCH_LOG_RTP_VALUES BUSINESS_RULE_LAUNCH_LOG_CALC_SCRIPT	See Setting Business Rules Launch Logging Properties .
CAPTURE_RTP_ON_JOB_CONSOLE	See About Runtime Prompts .
MAX_VALIDATION_RECORDS VALIDATION_CACHE_SIZE	See Setting Data Validation Properties .
MAX_CELL_TEXT, MAX_CELL_NOTE	See Setting the Maximum Length for Text Values and Comments in Cells .
DATA_GRID_CACHE_SIZE	See Specifying How Many Forms Users Can Simultaneously Use .
WARNING_THRESHOLD_NUM_OF_CELLS, ERROR_THRESHOLD_NUM_OF_CELLS, WARNING_THRESHOLD_NUM_OF_PUS, ERROR_THRESHOLD_NUM_OF_PUS, WARNING_THRESHOLD_FOR_NUM_DIMS, ERROR_THRESHOLD_FOR_NUM_DIMS, WARNING_THRESHOLD_FOR_NUM_ACCOUNTS, WARNING_THRESHOLD_FOR_NUM_ENTITIES, WARNING_THRESHOLD_FOR_NUM_VERSIONS	See Setting Threshold Limits for Applications .
, WARNING_THRESHOLD_FOR_NUM_SCENARIOS, WARNING_THRESHOLD_FOR_NUM_YEARS, WARNING_THRESHOLD_FOR_NUM_CURRENCIES, WARNING_THRESHOLD_FOR_NUM_PERIODS, WARNING_THRESHOLD_FOR_NUM_CUSTOM_DIM_MBRS, WARNING_SYNC_THRESHOLD_FOR_NUM_BLOCKS, ERROR_SYNC_THRESHOLD_FOR_NUM_BLOCKS, WARNING_ASYNC_THRESHOLD_FOR_NUM_BLOCKS, ERROR_ASYNC_THRESHOLD_FOR_NUM_BLOCKS, WARNING_SYNC_THRESHOLD_FOR_NUM_PASSES, ERROR_SYNC_THRESHOLD_FOR_NUM_PASSES, WARNING_ASYNC_THRESHOLD_FOR_NUM_PASSES, ERROR_ASYNC_THRESHOLD_FOR_NUM_PASSES, WARNING_THRESHOLD_FOR_CUBE_LINK_NUM_BLOCKS, ERROR_THRESHOLD_FOR_CUBE_LINK_NUM_BLOCKS	

 **Note:**

In some cases, additional properties may be visible in the **Properties** page. Because the following properties are required by Oracle Hyperion Planning, they should not be changed even if they are visible: `SYSTEM_DB_RELEASE` and `SYSTEM_DB_VERSION`.

To set Planning application or system properties:

1. Select **Administration**, then **Application**, and then **Properties**.
2. Select:
 - **Application Properties**: Set properties for the current application.
 - **System Properties**: Set properties for all Planning applications.
3. To:
 - Change a property, change its value under **Property Value**.
 - Add a property, click **Add**. Enter a name (avoid using spaces) and value for the property in the blank row.

If you enter a property already on the tab, its name turns red to indicate that it is a duplicate. You can duplicate properties between the application and system tabs.
4. Click **Save** and confirm your changes.
5. Stop and then restart the application server.

Properties are saved in the `HSPSYS_PROPERTIES` system database table, which by default is located in the relational database that you create when selecting the Configure Database task under Planning in EPM System Configurator.

Changing the JDBC Driver

By default, Oracle Hyperion Planning uses the embedded Oracle JDBC driver. To change the JDBC driver, update the JDBC property using these values:

Table 2-2 JDBC Property Values

JDBC Driver	RDB_SERVER_URL	RDB_DRIVER
Oracle Thin	<code>jdbc:oracle:thin:@%SERVER_NAME%:1521:%DB_NAME%</code>	<code>oracle.jdbc.driver.OracleDriver</code>
DB2 native	<code>jdbc:db2:%DB_NAME%</code>	<code>COM.ibm.db2.jdbc.app.DB2Driver</code>
DB2 native for remote DB2 server	<code>jdbc:db2://myhost.mydomain.com:6789/%DB_NAME%</code>	<code>COM.ibm.db2.jdbc.net.DB2Driver</code>

Database driver names and URLs are stored in the `HSPSYS_DATASOURCE` database table.

After changing JDBC properties, to make the necessary third-party files accessible to the Planning server:

1. Locate the `.jar` file (for Oracle) or `.zip` file (for DB2):

- For Oracle, find `classes12.jar` in the `OraHome/jdbc/lib` directory.
 - For DB2, find `db2java.jar` or `db2java.zip` in the `/Program Files/sqllib/java` directory.
2. Copy the `.jar` or `.zip` file from the previous step to the Web application server destination directory. For example:
 - a. Extract the `.ear` file.
 - b. Copy the `.jar` or `.zip` file to the `HPDomain/applications` directory.
 - c. Add the file to the `CLASSPATH` in `startHPServer.cmd`.
 3. Stop and restart the Web application server.

Configuring JDBC Connection Pooling

You can set the minimum and maximum number of JDBC connection pools. How you configure these depends largely on the number of Oracle Hyperion Planning users accessing the relational database. For example, you can specify the minimum connection number in the JDBC connection pool as one and the maximum as five. Doing so creates one connection when a user logs on to Planning. An additional connection is created for the next four users who log on to Planning, resulting in a total of five connections. Additional users who log on share the five connections.

The JDBC connection settings are set by default to a minimum of two and a maximum of ten. The Planning application does not function correctly if you reduce the maximum `JDBC_MAX_CONNECTIONS` to fewer than two.

To reconfigure JDBC connection pool parameters, change the minimum and maximum property values:

- `JDBC_MIN_CONNECTIONS`
- `JDBC_MAX_CONNECTIONS`

To optimize performance, Oracle recommends these settings:

- `JDBC_MIN_CONNECTIONS=10`
- `JDBC_MAX_CONNECTIONS=45`

Controlling Smart View Messages

If you use different releases of Oracle Smart View for Office and Oracle Hyperion Planning, a message is displayed when users start Smart View. To prevent the message from displaying, you can add the system property `SMART_VIEW_DISPLAY_WARNING`.

To control Smart View messages:

1. Select **Administration**, then **Application**, then **Properties**.
2. Select **System Properties** to set properties for all Planning applications.
3. Click **Add**, and enter this property in the blank row:
`SMART_VIEW_DISPLAY_WARNING`
4. Enter a value in **Property Value**:
 - **Yes**: Display messages.
 - **No**: Do not display messages.

5. Click **Save** and confirm your changes.

Showing New Planning Members in Smart View

You can import and edit Oracle Hyperion Planning metadata in Smart View. (See [Importing Planning Dimensions in Smart View](#).) New members created in the Oracle Smart View for Office grid are, by default, marked with an asterisk (*) in the grid. You can use a different sign by editing the Application Properties.

To show new Planning members with a sign other than an asterisk (*) in the Smart View grid:

1. Select **Administration**, then **Application**, then **Properties**.
2. Select **System Properties** to set properties for all Planning applications.
3. Click the SMART_VIEW_MD_NEW_MEMBER_SUFFIX.
4. Enter a value in **Property Value**:
5. Click **Save** and confirm your changes.

Setting the Application Home Page

You can specify the page that users see when they log in to a Oracle Hyperion Planning application so that they go directly to forms, task lists, or the Manage Approvals page. To do so, set the HOME_PAGE application property.

To set the application home page:

1. Select **Administration**, then **Application**, and then **Properties**.
2. Select **Application Properties** to set properties and values for a Planning application.
3. Add the property by clicking **Add**, entering the property in the blank row, and entering a value in **Property Value**:
 - Forms
 - TaskList
 - Approvals
4. Click **Save** and confirm your changes.

Setting Background Processing

You can set jobs—for business rules, Clear Cell Details, Copy Data, and Push Data—to switch to background processing after a threshold that you configure. After the specified period, jobs execute in the background. You can also set how often Oracle Hyperion Planning checks job status, displayed on the Job Console (see the *Oracle Hyperion Planning User's Guide*).

To set background processing and the frequency of status checks:

1. Select **Administration**, then **Application**, and then **Properties**.
2. Select **Application Properties** to set properties and values for a Planning application.
3. Add the property by clicking **Add**, entering the property in the blank row, and then entering a value in **Property Value**:

Table 2-3 Job Property Names and Values


PROPERTY_NAME Column	Property Value	Description
RULE_MAX_WAIT	The default and minimum value is 180,000 milliseconds (3 minutes). If you set a lower value, that value is ignored, and the property is set to the default value.	The interval in milliseconds to wait before running business rules in the background. If this property is not set, the feature does not work, and business rules do not run in the background.
		<div style="border: 1px solid #0070C0; padding: 10px; background-color: #E6F2FF;"> <p> Note:</p> <p>Business rules that are set to automatically run when a form is loaded or saved never run in the background.</p> </div>
RULE_MONITOR_DELAY	The default and minimum value is 180,000 milliseconds (3 minutes). If you set a lower value, that value is ignored, and the property is set to the default value.	The interval in milliseconds for checking the status of business rules.
CLR_CELL_MAX_WAIT	The default and minimum value is 180,000 milliseconds (3 minutes). If you set a lower value, that value is ignored, and the property is set to the default value.	The interval in milliseconds to wait before running Clear Cell Details jobs in the background. If this property is not set, Clear Cell Details jobs do not run in the background.
CLR_CELL_MONITOR_DELAY	The default and minimum value is 180,000 milliseconds (3 minutes). If you set a lower value, that value is ignored, and the property is set to the default value.	The interval in milliseconds for checking the status of Clear Cell Detail operations.
COPY_DATA_MAX_WAIT	The default and minimum value is 180000 milliseconds (3 minutes). If you set a lower value, that value is ignored, and the property is set to the default value.	The interval in milliseconds to wait before running Copy Data jobs in the background. If this property is not set, Copy Data jobs do not run in the background.
COPY_DATA_MONITOR_DELAY	The default and minimum value is 180,000 milliseconds (3 minutes). If you set a lower value, that value is ignored, and the property is set to the default value.	The interval in milliseconds for checking the status of Copy Data jobs.
PUSH_DATA_MONITOR_DELAY	The default and minimum value is 180,000 milliseconds (3 minutes). If you set a lower value, that value is ignored, and the property is set to the default value.	The interval in milliseconds for checking the status of Push Data jobs.
PUSH_DATA_MAX_WAIT	The default and minimum value is 180,000 milliseconds (3 minutes). If you set a lower value, that value is ignored, and the property is set to the default value.	The interval in milliseconds to wait before running Push Data jobs in the background. If this property is not set, Push Data jobs do not run in the background.

Table 2-3 (Cont.) Job Property Names and Values

PROPERTY_NAME Column	Property Value	Description
JOB_STATUS_MAX_AGE	Specify the value in milliseconds. The default value is 4 days (set in milliseconds). There is no minimum value.	<p>The maximum age of a completed job record before Planning deletes it from the database table, HSP_JOB_STATUS. Planning checks the job records every 30 minutes.</p> <p>For example, if you set the property value to 60,000 (1 minute), a job completes at 3:00, and Planning checks the job records at 3:01, then Planning would delete the completed job record.</p> <p>Removing completed job records can improve performance.</p>

4. Click **Save** and confirm your changes.

Setting Business Rules Launch Logging Properties

You can set application properties to enable logging when Oracle Hyperion Calculation Manager business rules are launched.



Note:

The logging levels specified in the ODL `logging.xml` file determine the level of messages that will be written to the logs. For information about ODL message types and logging levels, see the *Oracle Enterprise Performance Management System Installation and Configuration Troubleshooting Guide*.

These are the application properties that can be set to enable logging:

- **BUSINESS_RULE_LAUNCH_LOG_ENABLED:** Enables logging when Calculation Manager business rules are run and checks whether the next two properties are set to true. The default setting for this property is false.
- **BUSINESS_RULE_LAUNCH_LOG_RTP_VALUES:** If set to true, runtime prompt values are logged. The default setting for this property is false.
- **BUSINESS_RULE_LAUNCH_LOG_CALC_SCRIPT:** If set to true, the calculation script sent to Oracle Essbase is logged. The default setting for this property is false.

To set Calculation Manager business rules launch logging properties:

1. Select **Administration**, then **Application**, and then **Properties**.
2. Select **Application Properties** to set properties and values for a Oracle Hyperion Planning application.
3. Add the property by clicking **Add**, entering the property in the blank row, and entering a value (True or False) in **Property Value**:
 - BUSINESS_RULE_LAUNCH_LOG_ENABLED
 - BUSINESS_RULE_LAUNCH_LOG_RTP_VALUES
 - BUSINESS_RULE_LAUNCH_LOG_CALC_SCRIPT

4. Click **Save** and confirm your changes.

When these properties are set to enable logging, you will see the following files written to the log file directory:

- The `Planning_CalcLaunch.log` file logs information about start and stop times for the business rule launch, the user, runtime prompt values, and the calculation script.
- The `Planning_CalcExecution.log` file logs other exceptions that occur during execution of a business rule.
- The `Planning_CalcDeploy.log` file logs errors that occur during rule deployment.

Setting Data Validation Properties

You can set application properties for data validation rules in the application. With `MAX_VALIDATION_RECORDS`, you can specify the maximum number of validation failure entries to display in the **Data Validation Messages** pane that displays in forms. When the number of errors exceeds this limit, the entries for lower-priority rules are removed. Higher-priority errors remain, such as rules set to Do Not Promote. With `VALIDATION_CACHE_SIZE`, you can specify the maximum number of data validation rules that are cached to improve performance. For information on creating data validation rules, see [Managing Data Validation](#).

To set data validation properties:

1. Select **Administration**, then **Application**, and then **Properties**.
2. Select **Application Properties** to set properties and values for a Oracle Hyperion Planning application.
3. Add the property by clicking **Add**, entering the property in the blank row, and entering a value in **Property Value**:
 - `MAX_VALIDATION_RECORDS`: The maximum number of records to display in the **Data Validation Messages** pane. The default setting is 100.
 - `VALIDATION_CACHE_SIZE`: The maximum number of data validation rules that can be cached. The default setting is 10,000.
4. Click **Save** and confirm your changes.

Setting the Maximum Length for Text Values and Comments in Cells

Users can add cell text values and comments to form cells as described in the *Oracle Hyperion Planning User's Guide*. By default, the maximum number of single-byte characters allowed for text in each cell is 255, and the maximum number for comments is 1,500.



Note:

In the database, the column data type is set to `varchar(255)` and `varchar(2,000)` by default.

If you need to display additional characters in cell text or comments, you can set these Oracle Hyperion Planning application properties to the maximum length required by your application:

- `MAX_CELL_TEXT_SIZE`: Text values entered in cells whose data type is set to text
- `MAX_CELL_NOTE_SIZE`: Comments added to cells

Updating these settings to more than 2,000 requires that you make a corresponding change to the database. If you increase the maximum number of characters allowed for cell text, you must alter the database column size or type to support the changed size. (Changing the column type to CLOB, NCLOB, TEXT, or NTEXT to accommodate large cell text size can affect performance. Do so only if your application requires large cell text entries.) For additional information, see the documentation for your database.

To set the maximum length for text values and comments in cells:

1. Select **Administration**, then **Application**, and then **Properties**.
2. Select **Application Properties** to set properties and values for a Planning application.
3. Add the property by clicking **Add**, and enter one of these properties in the blank row:
 - `MAX_CELL_TEXT_SIZE`
 - `MAX_CELL_NOTE_SIZE`
4. Enter a value in **Property Value** to represent the maximum number of single-byte characters allowed for text values or comments in each cell.
5. **Optional:** If you are updating both of these properties, repeat step 3 and step 4 for the other property.
6. Click **Save** and confirm your changes.
7. Stop, and then restart the Planning server.
8. Back up the database, and then update the database column size or type to support the changed size specified in this property. For additional information, see the documentation for your database.

Controlling Smart View Cell Formatting in Planning Forms

Administrators can set the `SMART_VIEW_MERGE_FORMATTING` property to control whether cell formatting that administrators set in Oracle Smart View for Office merges with the cell formatting that non-administrators set in Smart View.

To use the `SMART_VIEW_MERGE_FORMATTING` property, add it using the instructions in [Setting Application and System Properties](#) and set its value:

- If set to `true`, in a Oracle Hyperion Planning form, the cell formatting that a non-administrator saved in Smart View is merged with the formatting that an administrator saved.
- If set to `false` (the default), in a Planning form, the cell formatting that a non-administrator saved in Smart View persists, even if an administrator changes the formatting of those cells.

For more information on how cell formatting applied in Smart View is applied to the same cells in Planning forms, see [How Cell Formatting in Smart View Persists in Planning](#).

Setting Threshold Limits for Applications

You can set threshold limits for applications—warning thresholds and error thresholds—that, if exceeded, will issue warnings in application monitor so artifacts can be modified or deny system requests at runtime.

You can set the application threshold limits by adding the application properties described in this section.

To set threshold limits for applications:

1. Select **Administration**, then **Application**, and then **Properties**.
2. Select **Application Properties** to set properties and values for a Oracle Hyperion Planning application.
3. Add the property by clicking **Add**, entering the property in the blank row, and then entering a value in **Property Value**:

Table 2-4 Threshold Property Names and Descriptions



PROPERTY_NAME Column	Description
<p>Thresholds for the number of cells in a data entry form:</p> <p>WARNING_THRESHOLD_NUM_OF_CELLS</p> <p>ERROR_THRESHOLD_NUM_OF_CELLS</p>	<p>A warning is issued and the artifact is shown in yellow in the application monitor if the number of cells in a form exceeds the warning threshold value.</p> <p>If the number of cells in a form exceeds the error threshold value, the form will not open.</p> <div style="border-left: 2px solid green; border-right: 2px solid green; border-bottom: 2px solid green; padding: 10px; margin-top: 10px;"> <p> Tip:</p> <p>Consider redesigning the form to reduce the number of cells by reducing the number of segments and/or members. Use of suppression options may temporarily reduce the total cells retrieved, but the total cells may exceed the threshold as data increases.</p> </div>
<p>Thresholds for the number of planning units in a planning unit hierarchy:</p> <p>WARNING_THRESHOLD_NUM_OF_PUS</p> <p>ERROR_THRESHOLD_NUM_OF_PUS</p>	<p>A warning is issued and the artifact is shown in yellow in the application monitor if the number of planning units in a planning unit hierarchy exceeds the warning threshold value.</p> <p>If the number of planning units in a planning unit hierarchy exceeds the error threshold value, you will not be able to save the planning unit hierarchy.</p> <div style="border-left: 2px solid green; border-right: 2px solid green; border-bottom: 2px solid green; padding: 10px; margin-top: 10px;"> <p> Tip:</p> <p>Consider redesigning the planning unit hierarchy to reduce the number of planning units by reducing the selected primary and secondary members.</p> </div>

Table 2-4 (Cont.) Threshold Property Names and Descriptions






PROPERTY_NAME Column	Description
<p>Thresholds for the number of dimensions in the application:</p> <p>WARNING_THRESHOLD_FOR_NUM_DIMS ERROR_THRESHOLD_FOR_NUM_DIMS</p>	<p>A warning or error is issued and the artifact is shown in yellow or red in the application monitor if the number of dimensions exceeds the warning or error threshold value, respectively.</p> <div data-bbox="950 415 1466 594" style="background-color: #e0ffe0; padding: 10px; border: 1px solid #800000;"> <p> Tip: Try reducing the number of dimensions.</p> </div>
<p>Thresholds for the number of members in each dimension:</p> <p>WARNING_THRESHOLD_FOR_NUM_ACCOUNTS WARNING_THRESHOLD_FOR_NUM_ENTITIES WARNING_THRESHOLD_FOR_NUM_VERSIONS WARNING_THRESHOLD_FOR_NUM_SCENARIOS WARNING_THRESHOLD_FOR_NUM_YEARS WARNING_THRESHOLD_FOR_NUM_CURRENCIES WARNING_THRESHOLD_FOR_NUM_PERIODS WARNING_THRESHOLD_FOR_NUM_CUSTOM_DIM_M BRS</p>	<p>A warning is issued and the artifact is shown in yellow in the application monitor if the number of members in a dimension (accounts, entities, versions, scenarios, and so on) exceeds the warning threshold value.</p> <div data-bbox="950 808 1466 1100" style="background-color: #e0ffe0; padding: 10px; border: 1px solid #800000;"> <p> Tip: To optimize calculation performance and data storage, try reducing the number of members in this dimension. Consider purging historical data (along with the corresponding members).</p> </div>
<p>Thresholds for business rules run in the context of a form:</p> <p>WARNING_SYNC_THRESHOLD_FOR_NUM_BLOCKS ERROR_SYNC_THRESHOLD_FOR_NUM_BLOCKS</p> <p>Thresholds for long running business rules run stand alone:</p> <p>WARNING_ASYNC_THRESHOLD_FOR_NUM_BLOCKS ERROR_ASYNC_THRESHOLD_FOR_NUM_BLOCKS</p>	<p>A warning or error is issued and the artifact is shown in yellow or red in the application monitor if the number of blocks processed in a rule exceeds the warning or error threshold value, respectively.</p> <div data-bbox="950 1316 1466 1520" style="background-color: #e0ffe0; padding: 10px; border: 1px solid #800000;"> <p> Tip: Consider adding additional scoping to the business rule to reduce the number of blocks processed.</p> </div>

Table 2-4 (Cont.) Threshold Property Names and Descriptions

PROPERTY_NAME Column	Description
<p>Thresholds for business rules run in the context of a form:</p> <p>WARNING_SYNC_THRESHOLD_FOR_NUM_PASSES</p> <p>ERROR_SYNC_THRESHOLD_FOR_NUM_PASSES</p> <p>Thresholds for long running business rules run stand alone:</p> <p>WARNING_ASYNC_THRESHOLD_FOR_NUM_PASSES</p> <p>ERROR_ASYNC_THRESHOLD_FOR_NUM_PASSES</p>	<p>A warning or error is issued and the artifact is shown in yellow or red in the application monitor if the number of passes made through the database in a rule exceeds the warning or error threshold value, respectively.</p> <div style="border: 1px solid green; background-color: #e0ffe0; padding: 10px; margin-top: 10px;"> <p> Tip:</p> <p>Consider grouping dimensions wherever possible when you calculate a series of dimensions. Avoid using parentheses unnecessarily when you calculate a series of formulas. You may also reconsider the usage for CALCMODE and CREATENONMISSINGBLK options in the calculation script.</p> </div>
<p>Thresholds for data export from a source during Reporting Mappings execution:</p> <p>WARNING_THRESHOLD_FOR_CUBE_LINK_NUM_BLOCKS</p> <p>ERROR_THRESHOLD_FOR_CUBE_LINK_NUM_BLOCKS</p>	<p>A warning is issued and the artifact is shown in yellow in the application monitor if the number of blocks from which the data is being exported exceeds the warning threshold value.</p> <p>If the number of blocks from which the data is being exported exceeds the error threshold value, Reporting Mappings will not execute at runtime.</p> <div style="border: 1px solid green; background-color: #e0ffe0; padding: 10px; margin-top: 10px;"> <p> Tip:</p> <p>Consider adding additional scope to the mapping to reduce the number of blocks processed.</p> </div>

4. Click **Save** and confirm your changes.

Logging On to EPM Workspace and Accessing Planning

You work with Oracle Hyperion Planning in the Oracle Hyperion Enterprise Performance Management Workspace environment. The default EPM Workspace URL is `http://web_server:port/workspace/`, where *web server* is the Web server machine hostname and *port* is the Web server port number, for example, 19000, if using the Oracle HTTP server instance configured by EPM System Configurator. Communicate the URL to all Planning users to enable them to log on to EPM Workspace and access Planning.

For information on installing and configuring EPM Workspace, see the *Oracle Enterprise Performance Management System Installation and Configuration Guide*.

To log on to EPM Workspace and access Planning:

1. Ensure that the Web server is started and the Web application server is running in the Services panel.
2. In your browser, enter the URL for the EPM Workspace **Log On** page.
3. Enter your user name.
4. Enter your password.
5. Click **Log On**.
6. In EPM Workspace, select **Navigate**, then **Applications**, and then **Planning**. Select a Planning application. If prompted, enter your logon information. Planning does not support non-ASCII characters for passwords.

You can log on to several Planning applications simultaneously, and navigate among them in the EPM Workspace tabs. The application names display as tabs at the top of the window, and you can click the tabs to move between applications. You can also have two views of the same application in the EPM Workspace tabs. If you want to open two or more instances of the browser to log on to EPM Workspace, you must append the EPM Workspace URL as described in the *Oracle Hyperion Enterprise Performance Management Workspace User's Guide*.

Planning supports users on a variety of network bandwidths. The performance on a 56K dialup connection can be enhanced by using HTTP compression on the Planning server.



Note:

Access Planning only through the EPM Workspace URL, as described in this procedure.

Working with Planning Utilities

Administrators can use the following utilities with Oracle Hyperion Planning.



Note:

Because these utilities require files that are on the Planning server, you must run the utilities from the Planning server, not from a client.

Table 2-5 Planning Utilities

Utility	Description	See
BroadcastMessage	Use broadcast messaging to communicate a text message to all Planning users currently logged on to an application.	Using Broadcast Messaging
CalcMgrCmdLineLauncher	Launch business rules created with Oracle Hyperion Calculation Manager.	Launching Business Rules With a Utility
CubeRefresh	Create and refresh application databases used to store data in Oracle Essbase for each plan type in the application.	Working with Applications that Use Planning Application Administration

Table 2-5 (Cont.) Planning Utilities

Utility	Description	See
DeleteSharedDescendant	Delete shared dimension members that are descendants of a given member.	Working with Applications that Use Planning Application Administration
ExportSecurity	Export Planning access permissions to a file, enabling you to export and import access permissions across applications.	Exporting Access Permissions
FormDefUtil	Move form definitions between Planning applications by exporting or importing to or from an XML file.	Importing and Exporting Form Definitions
HBRMigrateSecurity	Migrate launch access permissions on business rules and their projects from Oracle Hyperion Business Rules to Calculation Manager business rules in Planning.	Migrating Business Rule Security
HspUnlockApp	Clear all records in the HSP_LOCK table.	Unlocking Applications
ImportFormDefinition	Import a form definition from a text file into a Planning form.	Importing Form Definitions
ImportSecurity	Load access permissions for users or groups from a text file into Planning.	Importing Access Permissions
MaintenanceMode	Grant and withdraw access to Planning applications during maintenance.	Using the MaintenanceMode Utility to Limit Application Access
Outline Load	Load metadata and data for Planning applications.	Working with the Outline Load Utility
PasswordEncryption	Enable suppressing password prompts when running Planning utilities that prompt for passwords.	Suppressing Password Prompts in Planning Utilities
ProvisionUsers	Synchronize Planning users, groups, and roles in the Oracle Hyperion Shared Services Console with a Planning application and with Essbase.	Synchronizing Users With the Provision Users Utility
PushData	Schedule pushing data to a reporting application.	Pushing Data with a Utility
SortMember	Sort dimension members for Entity, Account, Scenario, Versions, and user-defined custom dimensions.	Working with Applications that Use Planning Application Administration
TaskListDefUtil	Move task list definitions between Planning applications by exporting or importing task list definitions to or from an XML file.	Importing and Exporting Task Lists

Suppressing Password Prompts in Planning Utilities

When running Oracle Hyperion Planning utilities that prompt for passwords, administrators can use an option to suppress password prompts, for example, when running utilities in batch mode. To enable suppressing password prompts, use the `PasswordEncryption` utility to create a file that stores a password in encrypted form. After the file is set up, you can run Planning utilities with the `[-f:passwordFile]` option as the first parameter in the command line to skip the password prompt and use the password from the encrypted file. Each password file contains one password, and is stored in the location that you specify when running the utility.

The `PasswordEncryption` utility uses a command line interface. By default, the utility is installed in the `/planning1` directory (for the full path, see [About EPM Oracle Instance](#)).

To enable suppressing password prompts in Planning utilities:

1. Enter this command from the `planning1` directory, where `passwordFile` is the full file path and file name for the password file:
 - **Windows:** `PasswordEncryption.cmd passwordFile`
 - **UNIX:** `PasswordEncryption.sh passwordFile`
2. When prompted, enter your password.

The masked password is encrypted and stored in the file and location specified in `passwordFile`. For other Planning utilities with password prompts, you can use `[-f:passwordFile]` as the first parameter in the command line to skip the prompt and use the encrypted password from the password file specified in `passwordFile`.

About Text Files for Planning Utilities

These Oracle Hyperion Planning utilities use text files that must be saved in UTF-8 format: `FormDefUtil`, `SampleApp_Data`, `TaskListDefUtil`, and the `Outline Load` utility. Other utilities do not use text files or do not require specific encoding.

For example, if you update the text file for the `FormDefUtil` utility in Notepad, the file is saved in the correct encoding format by default. If you change the Encoding option or create a text file with a different encoding format, such as ANSI or Unicode, the utility does not work correctly. When saving the text file, ensure that UTF-8 encoding is selected.

Running Planning Utilities on UNIX

To run Oracle Hyperion Planning utilities on UNIX, users must have appropriate privileges. For example, they need execution privileges for the directory from which utilities are run, and read and write access to the log directory.

About EPM Oracle Instance

The EPM Oracle instance is defined when Oracle Enterprise Performance Management System products are configured. This guide refers to the EPM Oracle instance location as `EPM_ORACLE_INSTANCE`. The default location for the EPM Oracle instance is `MIDDLEWARE_HOME/user_projects/epmsystem1`. By default, Oracle Hyperion Planning utilities are installed in the `EPM_ORACLE_INSTANCE/Planning/planning1` directory, and Planning `.log` files are written to the `EPM_ORACLE_INSTANCE/diagnostics/logs/planning` directory.

Debug messages for Planning utilities (for example, `ProvisionUsers`, `ImportSecurity`, and `ExportSecurity`) are logged to a log file called `PlanningCLU.log` that is generated under `C:\Oracle\Middleware\user_projects\epmsystem1\diagnostics\logs\planning`. The debug level can be modified in the `loggingCLU.xml` file located under `C:\Oracle\Middleware\user_projects\epmsystem1\Planning\planning1`.

Using Application Servers

You can access Oracle Hyperion Planning applications through application servers. You must register application servers on your workstation.

About Updating Instances and Clusters

When creating applications, you select a data source that is associated with an instance (also called a cluster). If necessary, you can update clusters using the EPM System Configurator. See "Planning Cluster Management" in the *Oracle Enterprise Performance Management System Installation and Configuration Guide*.

Closing Planning and Essbase

To close Oracle Essbase Server:

1. Maximize the Essbase window.
2. Enter **Quit**.

To log off Oracle Hyperion Planning Web pages select **File**, then **Log Off**. You return to the Logon page.

To close Planning on the Web, select **File**, then **Exit**.

3

Setting Up Access Permissions

Levels of Access Permissions

Setting access permissions to Oracle Hyperion Planning application elements prevents unauthorized users from viewing or changing data. You can set access permissions at these levels:

- Authentication of provisioned users and groups by an external user directory. See the *Oracle Enterprise Performance Management System User Security Administration Guide* and its information on Planning.
- Oracle Hyperion Shared Services roles that set access permissions for managing application groups, applications, dimensions, users, and groups. For example, users must have these Shared Services roles to perform the specified tasks:
 - Project Manager: Creates and manages application groups in Shared Services.
 - Provisioning Manager: Provisions users and groups to applications.
- Dimensions, including user-defined dimensions. Assign access permissions to members by selecting the dimension property **Apply Security**. If you omit or clear the **Apply Security** setting, all users can access the dimension's members.

By default, the Account, Entity, Scenario, Version, Year, Period, and Currency dimensions are enabled for access permissions.

- Users and groups, which can vary among applications. Assign access to Planning application artifacts by using **Assign Access**.

After updating access permissions, refresh the application to update Oracle Essbase security filters.

Planning Artifacts That Can Be Assigned Access

You can assign access permissions to:

- Scenario members
- Version members
- Account members
- Entity members
- User-defined custom dimension members
- Launch privileges to business rules
- Forms
- Form folders and business rule folders
- Task lists

When you change the user type for a user, the user has full read/write access to the application through third-party tools until you refresh the Oracle Essbase database. After the database is refreshed, appropriate access permissions are applied to the user.

Types of Access Permissions

Access permissions include read, write, and none. You can also set who can launch which Oracle Hyperion Calculation Manager business rules.

- **Launch:** Allow launch privileges

Note:

View user types have no write access to dimension members, so cannot launch business rules having runtime prompts that include members, dimensions, member ranges, or cross-dimension runtime prompt types. They can, however, launch business rules having runtime prompts of other types (for example, date type).

- **No Launch:** Disallow launch privileges.

Note:

If a user inherits Launch access permission to a business rule by belonging to a group, and is also assigned No Launch permissions by belonging to another group, the more restrictive No Launch assignment takes precedence.

You can specify access permission for individual users and each group. When you assign a user to a group, that user acquires the group's access permissions. If an individual's access permissions conflict with those of a group the user belongs to, user access permissions take precedence.

Inheriting Access Permissions

Inheritance determines the user or group's access permissions. You can specify an attribute that causes the children or descendants of that member to inherit its access permissions. Assigned access permissions take precedence over inherited access permissions. You can include or exclude the member from the access permissions setting.

Table 3-1 Options for Inheriting Access Permissions

Inheritance Option	Access Permission Assignment
Member	Only to the currently selected member
Children	To all children members in the level below the currently selected member
iChildren	To the currently selected member and all children members in the level below it
Descendant	To all descendant members below the currently selected member

Table 3-1 (Cont.) Options for Inheriting Access Permissions

Inheritance Option	Access Permission Assignment
Descendant	To the currently selected member and all descendant members below it

How access permissions are evaluated

When evaluating access permissions, Oracle Hyperion Planning gives precedence in this order:

1. Role-level security. Users with the Administrator role have access to all application elements (except the Mass Allocate role, which must be assigned to use the **Mass Allocate** feature).
2. Role-level security. Users with the Administrator role have access to all application elements.
3. For interactive users and planner user types, access permissions that are specifically assigned to users.
4. Access assignments that are acquired by belonging to a group.
5. Parent-level assignments (for example, to parent members or folders).

Enabling Access Permissions for Dimensions

Use the **Dimensions Property** tab to set access permissions to user-defined custom dimension members.

To enable access permissions for dimensions:

1. Select **Administration**, then **Manage**, and then **Dimensions**.
2. In **Dimension**, select the dimension.
3. Select **Actions**, and then select **Edit**.
4. In **Dimension Properties**, select **Apply Security**.

Note:

If you do not select this option, there is no security on the dimension, and users can access its members without restriction. See [Enabling Access Permissions for Dimensions](#).

5. Click **Save**.

Click **Reset** to revert to previously saved values.

Assigning Access to Members and Business Rules

Before you can assign access to members of user-defined custom dimensions, you must select the **Apply Security** check box on the dimension's **Property** tab. See [Enabling Access Permissions for Dimensions](#).

To assign access to members or business rules:

1. Select the member or business rule:
 - **For members:** Select **Administration**, then **Manage**, then **Dimensions**, and then select the dimension and member.
 - **For business rules:** Select **Administration**, and then **Business Rule Security**. Select the folder containing the business rules, and then select the business rules.
2. Select **Actions**, and then select **Assign Access**.
3. **Optional:** To migrate a user or group's changed identity or their position in the user directory from Oracle Hyperion Shared Services Console to Oracle Hyperion Planning, click **Migrate Identities**.

 **Note:**

Do not use the **Migrate Identities** button to migrate a user or group's changed identity or their position in the user directory from Shared Services Console to Planning. Instead, you must run the `UpdateUsers` command line utility.

4. **Optional:** To remove deprovisioned or deleted users or groups from the Planning database to conserve space, click **Remove Nonprovisioned Users/Groups**.
5. Add, edit, or remove access.
See [Adding, Editing, and Removing Access](#).

Adding, Editing, and Removing Access

You can specify which users and groups can access the selected member or business rule.

To assign, edit, and remove access permissions to members or business rules:

1. Select the member or business rule:
 - **For members:** Select **Administration**, then **Manage**, and then **Dimensions**. Then select the dimension and member.
 - **For Oracle Hyperion Calculation Manager business rules:** Select **Administration**, and then **Business Rule Security**. Select the folder containing the business rules, and then select the business rules.
2. Select **Actions**, and then select **Assign Access**.
3. **Optional:** To migrate a user or group's changed identity or their position in the user directory from Oracle Hyperion Shared Services Console to Oracle Hyperion Planning, click **Migrate Identities**.
4. **Optional:** To remove deprovisioned or deleted users or groups from the Planning database to conserve space, click **Remove Nonprovisioned Users/Groups**.
5. To add access:
 - a. Click **Add Access**.
 - b. Select the users and groups to access the selected member or business rule.
Click **Users** to display all user names; click **Groups** to display all groups.
 - If there are multiple pages of users or groups, type the page number to go to in **Page**, and click **Go**.

- Click **Start** or **End** to navigate to the first or last page.
- Click **Prev** or **Next** to move to the previous or next page.
- c. **Optional for members:** Select a relationship.
For example, select **Children** to assign access to the children of the selected member.
- d. Select an option:
 - Click **Launch** to allow the selected users and groups to launch the selected business rules.
 - Click **No Launch** to prevent the selected users and groups from launching the selected business rules.
- e. For the selected users or groups, select the access type and click **Add**.
- f. Click **Close**.
- 6. To modify access:
 - a. Click **Edit Access**.
 - b. For the selected member or business rule, select the access type for the displayed users or groups.
Click **Users** to display all user names; click **Groups** to display all groups.
For business rules only:
 - Click **Launch** to allow the selected users and groups to launch the selected business rules.
 - Click **No Launch** to prevent the selected users and groups from launching the selected business rules.
 - c. **Optional for members:** Select a relationship.
For example, select **Children** to assign access to children of the selected member.
 - d. Click **Set**.
 - e. Click **Close**.
- 7. To remove access:
 - a. Select the users and groups for whom to remove access to the selected member or business rule.
Click **Users** to display all user names; click **Groups** to display all groups.
 - b. Click **Remove Access**.
 - c. Click **OK**.
 - d. Click **Close**.

About Effective Access Permissions to Shared Members

You cannot assign access directly to a shared member. A shared member inherits access permissions from its base member, parent, or ancestor.

Oracle Hyperion Planning checks access permissions at each level, first by user, then by group, based on the member's access permissions inheritance relationship. If multiple access permissions exist, the least restrictive access permission is applied (for example, Write access takes precedence over Read access).

This example shows how effective access to base members and their shared members are determined if the database is refreshed or created with the **Security Filters** and **Shared Members** options selected in the **Refresh Database** or **Create Database** page (see [Creating and Refreshing Application Databases](#)).

Sample Entity Members

Parent Entity	Child Entities
United States	CA (base) NY
West	CA (shared) NV
Sales Region 1	CA (shared)

Table 3-2 Example of Inherited Access to Shared Members

Case	Access Permission	Effective Access for Base and Shared Member CA	Explanation
Case 1	CA (base) = None iDescendants (West) = Read	Read	CA inherits Read access from its West parent because Read is less restrictive than None.
Case 2	iDescendants (United States) = None iDescendants (West) = Read iDescendants (Sales Region 1) = Write	Write	CA inherits Write access from its Sales Region 1 parent because Write is less restrictive than Read or None.
Case 3	iDescendants (United States) = Write iDescendants (West) = None iDescendants (Sales Region 1) = Read	Write	CA inherits Write access from its United States parent because Write is less restrictive than Read or None.

Managing Access to Forms and Folders

Assigning Access to Forms and Folders

Administrators can assign access to forms, form folders, and Oracle Hyperion Calculation Manager business rule folders. (For information on assigning access to business rules and members, see [Types of Access Permissions](#) and [Assigning Access to Members and Business Rules](#).)

Principles:

- **Forms:**
 - Planners and interactive users can view or enter data only into forms to which they have access (and can work only with members to which they have access).

- Administrators and interactive users can design forms.
- Interactive users can access forms they created or to which an administrator assigned them access.
- Administrators have write access to all dimension members and to all forms.
- **Business rules:** Planners can see and launch only business rules to which they are assigned Launch access.
- **Form folders and business rule folders:**
 - Planners who are assigned access to a form folder can access the forms in that folder, unless they are assigned more specific access. Likewise, planners have Launch access to the Calculation Manager business rules in folders to which they are assigned access, unless they are assigned more specific access.
 - When you assign access to a folder, all folders under it inherit that access.
 - If you assign specific access (for example, None or Write) to a form folder, that access permission takes precedence over its parent folder's access permissions. For example, if a user has Write access to Folder1 that contains Folder2 to which the user has None access, the user can open Folder1, but does not see Folder2.
 - If you assign specific access (for example, Launch) to a Calculation Manager folder, that access permission takes precedence over its parent folder's access permissions. For example, if a user has Launch access to RulesFolder1 that contains RulesFolder2 to which the user has No Launch access, the user can open RulesFolder1, but does not see RulesFolder2.
 - If a user has None access to a form folder called Folder1 that contains a form called Form1 to which the user has Write access, the user can see Folder1 and Form1.
 - If a user has No Launch access to a Calculation Manager folder called RulesFolder1 that contains a business rule called Rule1 to which the user has Launch access, the user can see RulesFolder1 and Rule1.

For procedures, see [Adding, Changing, and Removing Access to Forms and Folders](#).

Adding, Changing, and Removing Access to Forms and Folders

To assign access to forms, form folders, and Oracle Hyperion Calculation Manager business rule folders:

1. Select the form or folder.
 - For forms and folders, see [Selecting and Opening Forms and Folders](#).
 - For business rule folders, select **Administration**, then **Manage**, and then **Business Rule Security**.
 - For business rules, select **Administration**, then **Manage**, and then **Business Rule Security**. Open the business rule folder containing the business rules and select the rules.

You can assign access to only one form, business rule, or folder at a time.

2. Select **Actions**, and then select **Assign Access**.
3. **Optional:** To migrate a user or group's changed identity or their position in the user directory from Oracle Hyperion Shared Services Console to Oracle Hyperion Planning, click **Migrate Identities**.

4. **Optional:** To remove deprovisioned or deleted users or groups from the Planning database to conserve space, click **Remove Nonprovisioned Users/Groups**.
5. To add access to forms and folders:
 - a. Click **Add Access**, and select the users or groups to access the form or folder.
Click **Users** to display all user names; click **Groups** to display all groups.
 - If there are multiple pages of users and groups, type the page number to go to in **Page**, and click **Go**.
 - Click **Start** or **End** to navigate to the first or last page.
 - Click **Prev** or **Next** to move to the previous or next page.
 - b. For **Type of Access**, select the kind of access users or groups have to the form or folder.
For business rules or their folders only:
 - Click **Launch** to allow the selected users and groups to launch the selected business rules.
 - Click **No Launch** to prevent the selected users and groups from launching the selected business rules.
 - c. Click **Add**.
 - d. Click **Close**.
6. To change which users can use or change forms or folders:
 - a. Select the users or groups for which to change access, and click **Edit Access**.
Click **Users** to display all user names; click **Groups** to display all groups.
 - b. For **Type of Access**, select the kind of access users or groups have to the form or folder.
 - c. Click **Set**.
 - d. Click **Close**.
7. To remove access from forms or folders:
 - a. Select the users or groups for which to remove access, and click **Remove Access**.
Click **Users** to display all user names; click **Groups** to display all groups.
 - b. Click **OK**.

Importing Access Permissions

The `ImportSecurity` utility loads access permissions for users or groups from a text file into Oracle Hyperion Planning. (To add users or groups, see the *Oracle Enterprise Performance Management System User Security Administration Guide*.) Importing access permissions overwrites existing access assignments only for imported members, forms, form folders, task lists, Oracle Hyperion Calculation Manager business rules, and Calculation Manager business rule folders. All other existing access permissions remain intact. The `SL_CLEARALL` parameter clears all existing access permissions; you can use it with other parameters to replace existing access permissions. See also [Exporting Access Permissions](#).

The `ImportSecurity` utility requires users to be provisioned to the Planning application before it assigns access. For example:

- If user `mrauch` is provisioned to the TotPlan application, this record will assign access permissions to `mrauch` successfully using the utility:

```
mrauch,member1,READWRITE,MEMBER
```

- If user ehennings is not already provisioned to the application, this record will fail to load:

```
ehennings,member1,READWRITE,MEMBER
```

The `ExportSecurity` utility automatically creates the `SecFile.txt` file, from which you can import access permissions. If you prefer, you can also manually create the `SecFile.txt` file using these guidelines:

- You must name the text file `SecFile.txt` and save it in the `planning1` directory (for the full path, see [About EPM Oracle Instance](#)).
- All users, groups, and artifacts must be defined in the application.
- Before importing access permissions on a user-defined custom dimension, you must allow access permissions to be set on it by selecting **Apply Security** (see [Enabling Access Permissions for Dimensions](#)).
- Each line in the `SecFile.txt` file must specify access permissions information.

Each line must contain these items, separated by one of these delimiters: comma (,), Tab, semi-colon (;), pipe (|), colon (:), space (.). Comma is the default.

Item	Description
<i>username or group name</i>	<p>The name of a user or group defined in Oracle Hyperion Shared Services Console.</p> <p>To import access permissions information into a group with the same name as a user, append this information to the line in the <code>SecFile.txt</code> file that pertains to the group: <code>sl_group</code></p> <p>For example:</p> <pre>admin,member1,READ,MEMBER admin,member1,READ,MEMBER,SL_GROUP</pre>
<i>artifact name</i>	<p>The named artifact for the imported access permissions (for example the member, form, task list, folder, or Calculation Manager business rule). Example: <code>Account1</code>.</p> <p>If an artifact name contains a character that you are using as the delimiter, enclose the name in double quotation marks. For example, if you are using a space as the delimiter, enclose the name South America in double quotation marks: <code>"South America"</code>.</p>
<i>access permissions</i>	<p>READ, READWRITE, or NONE. If there are duplicate lines for a user/member combination, the line with READWRITE access takes precedence. For example, for these lines:</p> <pre>User1,Member1,READ,@ICHILDREN User1,Member1,READWRITE,@ICHILDREN</pre> <p>Access permissions for User1 to Member1 are applied as READWRITE.</p> <p>For Calculation Manager business rules and folders only: specify launch access permissions as either NONE or LAUNCH.</p>

Item	Description
Oracle Essbase <i>access flags</i>	@CHILDREN, @ICHILDREN, @DESCENDANTS, @IDESCENDANTS and MEMBER. Security implementation for these functions is identical to Essbase.



Note:

For task lists, only MEMBER can be used. For folders, only @IDESCENDANTS can be used.

Item	Description
<i>artifact type</i>	<p>For artifacts other than members, distinguish which artifact you are importing security for with <i>artifact type</i> identifier:</p> <ul style="list-style-type: none">• SL_FORM—for forms• SL_COMPOSITE—for composite forms• SL_TASKLIST—for task lists• SL_CALCRULE—for Calculation Manager business rules• SL_FORMFOLDER—for form folders• SL_CALCOLDER—for folders containing Calculation Manager business rules

 **Note:**

The ExportSecurity utility automatically adds the required artifact type identifiers in the SecFile.txt file. If you manually create the SecFile.txt file, you must add the artifact type identifiers.

 **Note:**

The ExportSecurity utility does not support exporting access permissions to task lists for administrators, so you must manually add such records to the SecFile.txt file before you can import them.

Sample lines from a file:

```
User1,Account1,READ,@CHILDREN
```

```
Group2,DataForm08,READWRITE,MEMBER,SL_FORM
```

```
User3,TaskList09,READWRITE,MEMBER,SL_TASKLIST
```

```
NorthAmericaGroup,Sales,READWRITE,@IDESCENDANTS,SL_FORMFOLDER
```

To import access permissions into Planning:

1. Locate the `ImportSecurity` utility by navigating to the `planning1` directory (for the full path, see [About EPM Oracle Instance](#)).
2. From the Command Prompt, enter this case-sensitive command, one space, and the parameters, separating each with a comma. Enclose the parameters with double quotation marks:

```
ImportSecurity [-f:passwordFile] "appname,username,[delimiter],[RUN_SILENT],
[SL_CLEARALL]"
```

where:

Parameter	Description
<code>[-f:passwordFile]</code>	Optional: If an encrypted password file is set up, use as the first parameter in the command line to read the password from the full file path and name specified in <i>passwordFile</i> . See Suppressing Password Prompts in Planning Utilities .
<i>appname</i>	Name of the Planning application to which you are importing access permissions.
<i>username</i>	Planning administrator user name.
<i>delimiter</i>	Optional: <code>SL_TAB</code> , <code>SL_COMMA</code> , <code>SL_PIPE</code> , <code>SL_SPACE</code> , <code>SL_COLON</code> , <code>SL_SEMI-COLON</code> . If no delimiter is specified, comma is the default.
<code>RUN_SILENT</code>	Optional: Execute the utility silently (the default) or with progress messages. Specify 0 for messages, or 1 for no messages.
<code>[SL_CLEARALL]</code>	Optional: Clear existing access permissions when importing new access permissions. Must be in uppercase.

For example:

```
ImportSecurity "appl,admin,SL_TAB,1"
```

To clear all access permissions, enter:

```
ImportSecurity "appl,admin,,,SL_CLEARALL"
```

3. If prompted, enter your password.
4. After you execute the utility, check the log file `importsecurity.log` in the `EPM_ORACLE_INSTANCE/diagnostics/logs/planning` directory to verify the results. For the full path, see [About EPM Oracle Instance](#).

Improving Performance When Importing Access Permissions

To import access for many users, improve performance by not using full names:

1. On Oracle Hyperion Planning Web, select **Administration**, then **Application**, then **Settings**.
2. Select **Advanced Settings**, click **Go**, and then **System Settings**.
3. Clear **Display Users' Full Names**.

Exporting Access Permissions

The `ExportSecurity` utility exports Oracle Hyperion Planning access permissions to the `SecFile.txt` file, enabling you to export and import access permissions across applications (see [Importing Access Permissions](#)). For the specified user or group (or for all users and groups if you use only the mandatory parameters), the `ExportSecurity` utility exports access permissions to these artifacts: members, forms, form folders, task lists, business rules, and business rule folders. `ExportSecurity` appends an artifact type flag that specifies whether the exported artifact security is for a form, composite form, form folder, task list, business rule, or business rule folder.

Notes:

- If you specify only mandatory (not optional) parameters, all access permissions to all artifacts for all users and groups are exported. You can limit the export by specifying a member parameter (but only one member-based parameter).
- You can specify the optional parameters in any order.
- You can use only `/S_USER` or `/S_GROUP`, not both.
- Use the `/S=searchCriteria` parameter to specify users and groups with the same name.
- Running the utility creates a file named `SecFile.txt`, which contains the exported access permissions.

To export access permissions from Planning to a text file:

1. Navigate to the `planning1` directory (for the full path, see [About EPM Oracle Instance](#)).
2. From the Command Prompt, enter this case-sensitive command, one space, and the parameters. Separate each parameter with a comma:

```
ExportSecurity [-f:passwordFile] /A=appname,/U=username, [/S=searchCriteria|/
S_USER=user|/S_GROUP=group], [/S_MEMBER=memberName|/
S_MEMBER_ID=memberName |/S_MEMBER_D=memberName|/
S_MEMBER_IC=memberName|/S_MEMBER_C=memberName], [/DELIM=delim] , [/
DEBUG=true|false],[/TO_FILE=fileName] , [/HELP=Y]
```

where:

Parameter	Description	Mandatory?
<code>[-f:passwordFile]</code>	Optional: If an encrypted password file is set up, use as the first parameter in the command line to read the password from the full file path and name specified in <code>passwordFile</code> . See Suppressing Password Prompts in Planning Utilities .	No
<code>/A=appname</code>	The name of the Planning application from which you are exporting access permissions.	Yes
<code>/U=username</code>	The administrator's ID for logging into the application.	Yes
<code>/S=searchCriteria</code>	The user or group name. You cannot use this option with <code>/S_USER</code> or <code>/S_GROUP</code> .	No

Parameter	Description	Mandatory?
<code>/S_USER=<i>user</i></code>	A specified user name. You cannot specify multiple users or use this option with <code>/S_GROUP</code> or <code>/S=<i>searchCriteria</i></code> .	No
<code>/S_GROUP=<i>group</i></code>	A specified group. Only matching groups, not matching user names, are exported. You cannot specify multiple groups or use this option with <code>/S_USER</code> or <code>/S=<i>search criteria</i></code> .	No
<code>/S_MEMBER=<i>MemberName</i></code>	A specified member. You can specify only one member-based parameter.	No
<code>/S_MEMBER_ID=<i>MemberName</i></code>	A specified member and its descendants.	No
<code>/S_MEMBER_D=<i>MemberName</i></code>	A specified member's descendants.	No
<code>/S_MEMBER_IC=<i>MemberName</i></code>	A specified member and its children.	No
<code>/S_MEMBER_C=<i>MemberName</i></code>	A specified member's children.	No
<code>/DELIM=<i>delim</i></code>	SL_TAB, SL_COMMA, SL_PIPE, SL_SPACE, SL_COLON, SL_SEMI-COLON. If no delimiter is specified, comma is the default.	No
<code>/DEBUG=</code>	Specify <code>true</code> to display the utility's performed steps. <code>false</code> is the default.	No
<code>/TO_FILE=</code>	Specify the path to the <code>SecFile.txt</code> file. By default, the file is in the <code>planning1</code> directory (for the full path, see About EPM Oracle Instance). If you specify another path, use double backslashes, for example: <code>C:\Oracle\SecFile.txt</code> .	No
<code>/HELP=Y</code>	Specify as the only parameter to display the syntax and options for <code>ExportSecurity</code> .	No

For example, to export access permissions for a user and group named `Sales`, enter:

```
ExportSecurity /A=app1,/U=admin,/S=Sales
```

To export for a member named `Account100` and its descendants, with the colon delimiter to a file named `Account100.txt` in a specific path (in this example, to `Planning\planning1`):

```
ExportSecurity /A=planapp1,/U=admin,/TO_FILE=D:\\EPM_ORACLE_INSTANCE\
\Planning\planning1\Account100,/S_MEMBER_ID=Account100,/DELIM=SL_COLON
```

3. If prompted, enter your password.

Also note:

- If a member, user, or group name contains a character used as the delimiter, the name is enclosed in double quotation marks. For example, if a space is the delimiter, the name `South America` is enclosed in double quotation marks: `"South America"`.

- Because commas are used to separate parameters, if a parameter contains commas (for example, `Kravets, Diana`), precede it with a backslash. Also use backslash to escape the backslash from the command prompt. In this example, use two backslashes: `/A=Kravets\ \,Diana`
- The `ExportSecurity` utility does not support exporting access permissions to task lists for administrators, so you must manually add such records to the `SecFile.txt` file before you can import them.

Understanding the export file:

Item	Description
<i>user or group</i>	The name of a user or group defined in Oracle Hyperion Shared Services Console.
<i>memName</i>	A member in the application.
<i>access permissions</i>	<p>READ, READWRITE, or NONE. If there are duplicate lines for a user name/member name combination, the line with READWRITE access takes precedence.</p> <p>For Oracle Hyperion Calculation Manager business rules and folders only: Access permissions are specified as either NONE or LAUNCH.</p>
<i>Oracle Essbase access flags</i>	<p>@CHILDREN, @ICCHILDREN, @DESCENDANTS, @IDESCENDANTS, and MEMBER.</p> <p>Security implementation for these functions is identical to Essbase.</p>
<i>artifact type</i>	<p>After each line, the utility appends the artifact type:</p> <ul style="list-style-type: none"> • <code>SL_FORM</code>—for forms • <code>SL_COMPOSITE</code>—for composite forms • <code>SL_TASKLIST</code>—for task lists • <code>SL_CALCRULE</code>—for business rules • <code>SL_FORMFOLDER</code>—for form folders • <code>SL_CALCFolder</code>—for folders containing business rules



Note:

If you manually create the `SecFile.txt` file, you must add the artifact type identifiers.

For example, an exported file might contain these lines:

```
User1,DataForm2,READ,MEMBER,SL_COMPOSITE
User2,Folder3,READWRITE,MEMBER,SL_FORMFOLDER
User3,DataForm4,READWRITE,MEMBER,SL_FORM
```

"North America",Account101,READWRITE,MEMBER,SL_CALC_FOLDER

Reporting on Access Permissions

You can view current access permissions and print reports.

To report on current access permissions for users and groups in Oracle Hyperion Planning:

1. In Planning, select **Tools**, then **Reports**, and then **Access Control**.
2. On **Select User or Group**, select from the available options.
3. From the left **Available** panel, select and move users or groups to report to the **Selected** panel:

If you enter a user or group name instead of browsing to it, you must enter the full name. For names with commas, enclose the name in quotation marks.

4. Click **Next**.

Selecting Reporting Objects

You can report on these objects: Accounts, Scenarios, Versions, Entities, user-defined custom dimensions, and forms.

To select reporting objects:

1. Start the Access Control Report.
See [Reporting on Access Permissions](#).
2. On **Select Objects**, select the Oracle Hyperion Planning objects on which to report.
3. Click **Next**.

Selecting Reporting Options


To specify options for access reports:

1. Start the Access Control Report.
See [Reporting on Access Permissions](#).
2. On **Report Options**, for **Show Matching Access of Type**, select the access to view: **Read**, **Write**, or **None**.
3. For **Group the Results By**, select how to view the report: **Users** or **Objects**.
4. From the **Report Type** sections, select **Assigned Access** or **Effective Access**:

Table 3-3 Access Report Types

Report Type	Description	Options
Assigned Access	Summarizes access permissions that administrators assign	Specify whether access is assigned by member selection relation or group membership: <ul style="list-style-type: none"> • Show Matching Access of Relation: Member, Children, Children (inclusive), Descendants, or Descendants (inclusive). • Show Inherited From Group: Show access permissions inherited by users in a group.
Effective Access	Summarizes access assignments as Oracle Hyperion Planning evaluates them (for example, by member selection relation, such as children, or group membership). This is useful if there are conflicts in access permissions.	Describe the origin of the effective access by selecting Show Effective Access Origin . For example, a user named JSomebody may be assigned Write access to Entity1 and belong to a group named Sales that is assigned Read access to Entity1. This setting would show that JSomebody has Write access to Entity1 because individual assigned access supersedes access inherited by group membership.

Effective Access

 **Note:**
Effective Access reports cannot be generated for groups.

5. Click Finish.

Adobe Acrobat launches, displaying the report online.

Working with Access Permissions Reports

The report on access permissions displays in Adobe Acrobat. You can use the Adobe Acrobat toolbar to work with the report.

Setting Up Audit Trails

Administrators can select aspects of the application for change tracking. For example, you can track changes to metadata, such as when users change a member property or add a currency. You can also track changes in forms, business rules, approvals, users, access permissions, and so on. To view audits, administrators create and run reports using RDBMS report writers.

Access assignments imported with the `ImportSecurity` utility are not reflected in audit reports.

Table 3-4 Actions That Can be Audited

Audit Options	Tracked Changes
Dimension Administration	<ul style="list-style-type: none"> • Dimension hierarchy: Adding a member or dimension; moving, deleting, or changing properties; renaming a member or dimension • Performance settings: Resetting a dimension's dense or sparse setting, changing the order of dimensions • Currencies: Adding or deleting currencies, setting a triangulation or reporting currency • Updates by utilities that affect Oracle Hyperion Planning (such as importing form designs with the <code>ImportFormDefinition</code> utility)
Alias Table Administration	Changes to alias tables: Creating, copying, renaming, deleting, and clearing
Data	<ul style="list-style-type: none"> • Cell values • Supporting detail • Account annotations • Cell-level documents
Launch Business Rules	Updates from calculation scripts and business rules (including runtime prompts)
Form Definition	Forms: Creating, modifying, adding rows. (The audit record does not record how the design changed.)
Form Folder Administration	Folders: Created, moved, or deleted
Approvals	Approvals: Planning unit owners, status, and status (started or excluded)
Copy Version	Versions copied, including supporting detail and annotations. The audit record does not record details (such as data, supporting detail, and annotations) of the copied version.
Security	Access permissions to dimension members, forms, form folders, business rules, and task lists
Users Administration	Users added, changed, or deleted
Groups Administration	Groups added, changed, or deleted; users added or removed
Task List	Task lists: created, copied, saved, moved, and deleted
Copy Data	Users' selections for Static Dimensions, Source Dimension, and Destination Dimension
Clear Cell Details	Users' selections for clearing supporting detail, cell text, and account annotations

To specify aspects of the application for which Planning records changes:

1. Select **Tools**, and then **Reports**.
2. Select **Auditing**.
3. Select the actions Planning tracks.

To avoid affecting performance, be selective in which application elements you audit.

4. Click Save Selections.

Depending on selected audit options, application changes are recorded in a `HSP_AUDIT_RECORDS` table, stored in the relational database.

5. Restart the application server.**6. View results in the `HSP_AUDIT_RECORDS` table using a RDBMS report writer.**

If anyone resets audit options, those changes are recorded.

Examples of Tracked Actions

For each recorded action, Oracle Hyperion Planning tracks:

Tracked Changes	Examples
The type of change	Metadata, data, form, access permissions, planning units
The affected object (The columns <code>ID_1</code> and <code>ID_2</code> in the audit report help define the object that changed.)	Form: Expenses 04 Group: Marketing
User	VHennings
Time Posted	08/22/2013 8:17
Action	Add
Property	Currency
Old value	Default
New value	USD

Viewing and Clearing Audit Reports

Audit results are recorded in the `HSP_AUDIT_RECORDS` table, stored in the relational database. To clear the audit report, use the SQL `DELETE` command on the `HSP_AUDIT_RECORDS` table. To clear entries that are a certain number of days old, compare them against the `time_posted` field. For example, to delete all entries from the table:

```
DELETE FROM HSP_AUDIT_RECORDS
```

To view audit records, sorted by the time they were posted:

```
SELECT * FROM HSP_AUDIT_RECORDS ORDER BY TIME_POSTED
```

Managing Security Filters

Access permissions in Oracle Hyperion Planning are stored in the relational database.

If you use other products outside of Planning, such as Oracle Hyperion Financial Reporting or third-party tools, to access Planning data directly in Oracle Essbase, you must push Planning access permissions to Essbase by generating security filters.

To update security filters for selected users, select **Administration**, then **Manage**, and then **Security Filters**. To update security filters simultaneously for all users, select **Administration**, then **Application**, then **Create Database** or **Refresh Database**, and then **Security Filters** (see [Creating and Refreshing Application Databases](#)).

First validate that the size of security filters does not exceed the Essbase limit of 64 KB per row.

 **Note:**

Oracle recommends that you perform a database refresh with the **Security Filters** option selected each time a change is made to the outline, including when dynamic child members are added or members are renamed and security is assigned to the new members.

For read and write security filters to be generated in Essbase, users must have read or write access permissions to at least to one member from each secured Planning dimension, including user-defined dimensions. If access is not assigned in these dimensions, the security filter for the user in Essbase is set to None.

To create or update individual security filters:

1. From Planning, select **Administration**, then **Manage**, then **Security Filters**.
2. Select the users whose security filters you want updated.
3. Click **Create**.

Essbase creates an encrypted file (`essbase.sec`) to store access permissions information.

Notes:

- If you want planners and interactive user types to have write access directly to Planning data in Essbase, assign them the role "Essbase Write Access" in Oracle Hyperion Shared Services.
- After creating or refreshing security filters for a user who has access to dynamic members, the Essbase bucket name is displayed in the filter line for dynamic members instead of the actual member name.

Synchronizing Users With the Provision Users Utility

The `ProvisionUsers` utility—run by administrators through a command line interface—synchronizes Oracle Hyperion Planning users, groups, and roles in Oracle Hyperion Shared Services Console with a Planning application and with Oracle Essbase.

To use the utility:

1. Launch the `ProvisionUsers.cmd` file from the `planning1` directory, using the following syntax:

```
ProvisionUsers [-f:passwordFile] /ADMIN:adminName /A:appName [/U:user1[;user2;user3] [/R:n]
```

For the full path to `planning1`, see [About EPM Oracle Instance](#).

Table 3-5 ProvisionUsers Syntax

Parameter	Description	Required?
<code>[-f:passwordFile]</code>	Optional: If an encrypted password file is set up, use as the first parameter in the command line to read the password from the full file path and name specified in <i>passwordFile</i> . See Suppressing Password Prompts in Planning Utilities .	No
<code>/ADMIN:adminName</code>	The administrator's name for logging on to the Planning application.	Yes
<code>/A:appName</code>	The Planning application to synchronize (must be on the server on which the utility is run).	Yes
<code>[/U:user1[;user2;user3]]</code>	Specifies users to synchronize. For example, to synchronize users Planner1 and Planner2, use <code>/U:Planner1;Planner2</code> . Omitting this argument synchronizes all users.	No
<code>[/R:n]</code>	Specifies an interval, in minutes, in which synchronization is run. For example, to synchronize every 30 minutes, use <code>/R:30</code> . Omitting this argument performs the synchronization once.	No
<code>/?</code>	Specified by itself, prints the syntax and options for <code>ProvisionUsers</code> .	No

2. If prompted, enter your password.

Example 1

Entering:

```
ProvisionUsers /ADMIN:admin /A:App1
```

Synchronizes all users in the App1 application.

Example 2

Entering:

```
ProvisionUsers /ADMIN:admin /A:App2 /U:Planner1 /R:60
```

Synchronizes user Planner1 in the App2 application every 60 minutes.

Migrating User and Group Identities

When you change a user or group's identity or their position in the user directory hierarchy, you must update—or migrate—this information to Oracle Hyperion Planning.

To migrate changed user and group identities from Oracle Hyperion Shared Services Console to Planning:

1. Take an action:
 - Select **Administration**, then **Manage**, then **Dimensions**, and then select a dimension member.

- Select **Administration**, then **Manage**, then **Forms and Ad Hoc Grids**, and then select a form.
 - **If using Oracle Hyperion Calculation Manager:** Select **Administration**, then **Business Rule Security**, and then select a business rule folder or business rule.
 - Select **Administration**, then **Manage**, then **Task Lists**, and then select a task list.
2. Click **Assign Access**.
 3. Click **Migrate Identities**.

Migrating Business Rule Security

If your upgraded application used Oracle Hyperion Business Rules, administrators can migrate launch access permissions on business rules and their projects from Business Rules to Oracle Hyperion Calculation Manager business rules in Oracle Hyperion Planning using the `HBRMigrateSecurity.cmd` utility.

The `HBRMigrateSecurity.cmd` utility:

- Overwrites launch access permissions that are already assigned to business rules in the specified Planning application.
- Migrates access permissions only for users and groups that are provisioned for the specified Planning application in Oracle Hyperion Shared Services Console.



Note:

Because Business Rules is no longer supported in Planning and Calculation Manager is the only option, the **Calculation Module** setting was removed from the **System Settings** page.

To migrate access permissions on business rules and their folders:

1. Before running `HBRMigrateSecurity.cmd`:
 - Migrate business rules from Business Rules to Calculation Manager. See the *Oracle Hyperion Calculation Manager Designer's Guide*.
 - Deploy the business rules to Planning.
2. At the command line, from the `planning1` directory, enter this command and its parameters, separating each by a space:

```
HBRMigrateSecurity.cmd [-f:passwordFile] /A:appname /U:admin /F:output file
```

For the full path to `planning1`, see [About EPM Oracle Instance](#).

Table 3-6 HBRMigrateSecurity Parameters

Parameter	Purpose	Required?
<code>[-f:passwordFile]</code>	Optional: If an encrypted password file is set up, use as the first parameter in the command line to read the password from the full file path and name specified in <i>passwordFile</i> . See Suppressing Password Prompts in Planning Utilities .	No
<code>/A: appname</code>	Specify the Planning application to which to migrate launch access permissions for business rules	Yes
<code>/U: admin</code>	Specify the administrator's user name	Yes
<code>/F: output file</code>	Specify the name of the XML output file, including its full path if it is not in the <code>planning1</code> directory (for the full path to <code>planning1</code> , see About EPM Oracle Instance). This file contains a log of the transaction, and helps with troubleshooting.	Yes
<code>/?</code>	Print the syntax and options for <code>HBRMigrateSecurity.cmd</code>	No

3. If prompted, enter your password.

For example:

```
HBRMigrateSecurity.cmd /A:appname /U:admin /F:C:\temp\HBRExportedSecurity.xml
```

For information on the **Security** setting in Calculation Manager for migrated business rules, see [About Runtime Prompts and Approvals Security](#).

Removing Stale User Records

When you deprovision or delete users or groups in Oracle Hyperion Shared Services, you can conserve disk space by updating the user and group tables in the Oracle Hyperion Planning relational database to remove stale records.

To remove stale user records from the Planning database tables:

1. Take an action:
 - Select **Administration**, then **Manage**, then **Dimensions**, and then select a dimension member.
 - Select **Administration**, then **Manage**, then **Forms and Ad Hoc Grids**, and then select a form folder or form.
 - **If using Oracle Hyperion Calculation Manager:** Select **Administration**, then **Business Rule Security**, and then select a business rule folder or business rule.
 - Select **Administration**, then **Manage**, then **Task Lists**, and then select a task list.
2. Click **Assign Access**.
3. Click **Remove Nonprovisioned Users/Groups**.

Setting Up Access Permissions in Financial Reporting

Oracle Hyperion Financial Reporting supports these access permissions:

- User authentication
 - Logon access permissions
 - Access to Financial Reporting and data source
- Application permissions
 - Access to tasks within Financial Reporting
 - Permissions to design or view reports
- Data Rights
 - Access to data source data such as members and values
 - Access to Financial Reporting objects such as reports

4

Managing Planning Databases

Unlocking Applications

Occasionally Oracle Hyperion Planning applications can become locked, for example, if users abnormally exit the application and Planning. The Unlock Application utility clears all records in the `HSP_LOCK` table. You must run the utility from the Planning application server.

Ensure there are no users connected to the Planning application before running the utility. To confirm this, launch the task manager on the Planning server and ensure there are no processes called `hsxser~1` (`hsxserver`) or `hspds`.

To unlock Planning applications:

1. Locate the `HspUnlockApp.cmd` utility by navigating to the `planning1` directory using the command line.

For the full path to `planning1`, see [About EPM Oracle Instance](#).

2. Enter `HspUnlockApp.cmd[-f:passwordFile] SERVER_NAME USER_NAME PASSWORD APPLICATION_NAME`, where *application name* is the application to unlock.

Optional: If an encrypted password file is set up, use `[-f:passwordFile]` as the first parameter in the command line to read the password from the full file path and name specified in `passwordFile`. See [Suppressing Password Prompts in Planning Utilities](#).

3. If prompted, enter your password.
4. Check the application event logs using the Event Viewer in the console application log to determine whether a success or failure event is reported. For information about logs, see the *Oracle Enterprise Performance Management System Installation and Configuration Troubleshooting Guide*.

Using Broadcast Messaging

Use broadcast messaging to communicate a text message to Oracle Hyperion Planning users logged on to an application. For example, you can send messages about system availability or periodic maintenance. You should also send broadcast messages to request that users log out before upgrading or migrating applications.

You can send broadcast messages using the Web client. If you send them using the Web, they are sent to users of your current application. You can also schedule messages using standard operating system mechanisms. You can also send broadcast messages using a command line utility. If you send them using the command line, you can specify any application, without being logged on to it.

The messages are displayed in logged-on users' browsers upon page refresh or when they visit another page.

Users who are logged on to the application through other products or third-party reporting tools do not receive broadcast messages.

To send broadcast messages:

1. Select **Administration**, then **Application**, then **Broadcast Messages**.
2. In **Create Message**, enter the message to broadcast.
3. Click **Send**.

To send broadcast messages using a command line:

1. Locate the `BroadcastMessage.cmd` utility by navigating to the `planning1` directory with a command prompt.

For the full path to `planning1`, see [About EPM Oracle Instance](#).

2. Launch the utility, using this syntax at the command line:

```
broadcastmessage.cmd ([SERVER_NAME], APPLICATION_NAME, USER_NAME, MESSAGE)
```

Optional: If an encrypted password file is set up, use `[-f:passwordFile]` as the first parameter in the command line to read the password from the full file path and name specified in `passwordFile`. See [Suppressing Password Prompts in Planning Utilities](#).

SERVER_NAME: the localhost name.

APPLICATION_NAME: the name of the application to whose users you send messages.

USER_NAME: the administrator who has rights to send broadcast messages.

MESSAGE: a text message of up to 127 characters to send to application users.

3. If prompted, enter your password.
4. You can view status, errors, or information for broadcasting messaging in the console.

For example:

```
Broadcastmessage.cmd ABCserver, testapp, VHennings001, Please log off the application for routine maintenance.
```

Viewing Usage Statistics

You can determine which Oracle Hyperion Planning users are logged on to the current application, and how long since they accessed the application. Users are not listed if they log on through other applications such as Oracle Hyperion Financial Reporting, Oracle Smart View for Office, or third-party reporting tools.

You can view the percentage of supporting detail detection cache being used, to determine whether an appropriate amount of RAM is allocated. If the number is very low or high, consider allocating less or more RAM. A default value of 20 is stored as a Planning property (see [Allocating Memory for Supporting Detail Cache](#)).

To view usage statistics:

1. From Planning Web, log on to an application.
2. Select **Administration**, then **Application**, then **Statistics**.

Creating and Refreshing Application Databases

On the **Manage Database** page, you can create and refresh application databases, which are used to store data in Oracle Hyperion Planning for each plan type in the application.

Databases are structured according to dimensions, hierarchical members, attributes, and other data specified in an application.

Oracle Essbase creates an encrypted data file (`essbase.sec`) to store access permission information.

When you create an application, select **Create** to update the Planning multidimensional databases that store application data. While building your outline, you can transfer database changes and access permissions separately to improve performance and make changes quickly available to users. When the outline is complete, Oracle recommends that you include access permissions when refreshing database information.

You must refresh the application database whenever you change the application structure. Changes made to an application are not reflected to users performing data entry and approvals tasks until you refresh the Planning databases for the application. For example, when you modify properties of an Entity member, add a Scenario, or change access permissions, these changes are stored in the Planning relational database until you refresh the application database.

During refresh:

- Security filters are updated.
- Currency conversion calculation scripts are updated.
- Members and associated properties are propagated from the relational database to the multidimensional database.
- Custom attributes are added, modified, or deleted in the multidimensional database.
- Exchange rate values are repopulated in the Planning outline.
- Member formulas for certain accounts are dynamically generated or updated.
- Additions or changes to alias tables and their associations to dimensions or members are updated.
- The Planning application is restructured.
- UDAs are added to the Planning application.

 **Caution:**

Oracle recommends backing up the application before creating or refreshing. See [Backing Up Applications and Application Databases](#). Following these steps affects data in the database. When you click **Create**, data is erased and Planning plan types are rebuilt. When you click **Refresh**, data might be replaced. For important information, see [Considerations for Working with Essbase](#).

When an administrator uses **Create Database** or **Refresh Database**, all other tasks are unavailable to other users, including the application owner.

All users must be logged off from the Planning application before Planning databases can be refreshed. Oracle recommends that administrators send a broadcast message to all users, asking them to save their work and close the application before the Planning application is refreshed. Planning does not log out users during refresh. See [Limiting Use of an Application During Maintenance](#).

Application refresh time depends on factors such as the number of entities and users in the application, and many users and security filters lengthens refresh time. To maximize system availability, you can transfer access permissions information during nonpeak hours.

To create or refresh the application database:

1. Back up the application. See [Backing Up Applications and Application Databases](#).
2. Select **Administration**, and then **Application**.
3. Select **Create Database** or **Refresh Database**.
4. Select options:

- **Database:** Creates or refreshes a Planning database for the application.
- **Update custom-defined functions:** Updates Planning custom-defined functions for the application when creating or refreshing the application database.

For information about working with custom-defined functions, see the *Oracle Essbase Database Administrator's Guide*.

- **Security Filters:** Generates security filters for use by third-party applications. To generate security filters for all users in the application, select **Security Filters**, but do not select **Validate Limit**. To generate security filters for selected users, see [Managing Security Filters](#).

Access permissions are stored in an encrypted data file (`essbase.sec`).

 **Tip:**

Before you generate security filters for all users, limit user access to the application by setting the **Enable Use of the Application For** option to **Administrator**. After generating security filters, change the setting back to **All Users**. See [Limiting Use of an Application During Maintenance](#).

 **Note:**

After creating or refreshing security filters for a user who has access to dynamic members, the Essbase bucket name is displayed in the filter line for dynamic members instead of the actual member name.

- **Shared Members:** Evaluates access permissions that are set for all instances of the member (base and shared) and applies the least restrictive access permission to them all. For example, if a shared member's parent has Write access assigned to all its children, and another shared member under another parent has Read access assigned to all its children, the base member and all its shared members are given Write access (for examples, see [About Effective Access Permissions to Shared Members](#)).

 **Note:**

If your application does not depend on shared member security, consider clearing this option to improve performance.

If this option is cleared, shared members inherit the security assigned to the base member.

- **Validate Limit:** Identifies security filters that exceed the Essbase security filter limit of 64 KB per row. This option validates filter size to ensure it does not exceed the size limit before building Essbase security filters.
5. To create or refresh data in the Planning database, click **Create** or **Refresh**.
 6. Review the confirmation message. To continue, click **Create** or **Refresh**. After the update completes, click **Finish**.

Note:

If the create or refresh process takes some time, click **Run in Background** to run the process in the background without displaying the status.

To view the results of running **Create** or **Refresh** in the background, view the Planning logs. For information about logs, see the *Oracle Enterprise Performance Management System Installation and Configuration Troubleshooting Guide*.

Managing Exchange Rates

Exchange rate tables enable budget preparers to create plans in different currencies. For example, you can specify Yen as the base currency for the Japan entity and US dollars for the United States entity. When you display a form with values for the Japan entity and display currency set to US dollars, the Yen exchange rate is used to convert values for Japan to US dollars. If the display currency is set to Yen, the exchange rate for US dollars converts values for the United States entity to Yen.

To use exchange rates, you must select Multiple Currencies when creating an application. You can set up exchange rates using the procedure in [Creating Exchange Rate Tables](#) and [Editing Exchange Rate Tables](#).

A multiple currency application stores exchange rates with the HSP_Rates dimension, which includes these members and others that store currency rates:

- HSP_InputValue: Stores data values
- HSP_InputCurrency: Stores currency types for data values

When generating reports or loading data, you must refer to the HSP_InputValue member. When loading data, you must load data against the local currency. You need not refer to the HSP_InputCurrency member. By default, the HSP_Rates dimension is set to Sparse.

Creating Exchange Rate Tables

You can create multiple exchange rate tables, each representing a different business scenario. Each scenario can be associated with only one exchange rate table.

To create exchange rate tables:

1. Select **Administration**, then **Manage**, and then **Exchange Rates**.
2. Click **Create**.
3. Specify information for the Exchange Rate table.

- Click **Save**, and then define settings for the table as specified in [Editing Exchange Rate Tables](#).

Editing Exchange Rate Tables

The default currency and triangulation currencies are available as destination currencies. You can enter exchange rates from source currencies to the default or triangulation currencies. You enter conversion values between the default currency and all the currencies defined in the Exchange Rates page. Exchange rate tables span all application time periods, so you can apply exchange rates to all scenarios. When creating or modifying exchange rate tables, you must refresh the application to store them in the plan types.

If you modify a currency's triangulation currency, you must re-enter exchange rates for the triangulation currency property and refresh the application to transfer and store the exchange rates. You cannot select the application's default currency as a triangulation currency.

When you input exchange rates for converting from one currency to another, you can select Multiply or Divide as the calculation method.

To edit exchange rate tables:

- Select **Administration**, then **Manage**, then **Exchange Rates**, select the table to edit, then click **Edit**.
- In the **Rate Table** tab, select options.
- Click **Next**.
- On the **Exchange Rate** tab, set options:

Table 4-1 Exchange Rate Table Options

Option	Description
To Currency	The currency for which to enter conversion rates (the default currency or a triangulation currency).
Show Years	The time periods displayed (by default, the current application year).
Rate Table Name	The name of the exchange rate table (display only).
Method	Multiply or Divide, the mathematical operator that determines how values are calculated between the source and destination currencies.
Historical	For all time periods, the exchange rate for accounts whose Exchange Rate Type is set to Historical. The account's Data Type must be set to Currency. Historical is typically used for balance sheet account types. A historical exchange rate may reflect a calculated rate over time, a rate for a point in time before the application's calendar, or a rate that was in effect when an event occurred.

Table 4-1 (Cont.) Exchange Rate Table Options

Option	Description
BegBalance	The value of balance sheet accounts. There is one beginning balance time period, the first time period in the application. Rates for the Beginning Balance time period are populated for each year in the application. Scenarios that do not include the first year of the application can include a Beginning Balance time period.
Avg	For time periods, the exchange rate for accounts whose Exchange Rate Type is set to Avg, or Average. Avg is typically used for Revenue and Expense account types, or for Saved Assumption account types whose Time Balance is set to Flow. The account's Data Type must be Currency.
End	For time periods, the exchange rate for accounts whose Exchange Rate Type is set to Ending. Ending is typically used for Asset and Liability account types, or for Saved Assumption account types whose Time Balance is set to Balance. The account's Data Type must be Currency.



Tip:

After entering values, fill in the value throughout the current year or all years in the table. For example, if you enter a value for Avg in the Jan11 cell and select **Fill Year**, the value is spread to all the months in 2011. If you select **Fill Table**, the value is spread to the months for all the years included in this exchange rate table. To fill in values, enter a value for Avg or End, right-click the cell, and select **Fill Year** or **Fill Table**.

5. Click **Save**.

Deleting Exchange Rate Tables

To delete exchange rate tables:


1. Select **Administration**, then **Manage**, and then **Currency Conversion**.
2. Select the exchange rate table to delete.
3. Click **Delete**.
4. At the prompt, click **OK**.

Managing Currency Conversions

To manage currency conversions:

1. Select **Administration**, then **Manage**, and then **Currency Conversions**.
2. Click **Create**.

3. In the **Create File** tab, specify information for the Currency Conversion Script file, and then click **Next**.
4. In the **Details** tab, select information for the Currency Conversion Script details.

Click  to select members for fields.

Working with Currency Conversion Calculation Scripts

If multiple currencies are enabled for the Oracle Hyperion Planning application when you create a currency conversion, a currency conversion calculation script is created, based on selected scenarios, versions, and currencies. A second calculation script is generated by Planning, which copies appropriate exchange rates to the account, based on account rate types. For currency conversion, the Account type always takes precedence. Data type evaluation order is not considered. The copy calculation script is named `HspCrtB.csc` for bottom-up versions and `HspCrtT.csc` for target versions. Running the copy calculation script enables the currency conversion calculation script to run in BLOCK mode, which is more efficient than CELL mode.

The selected scenarios, versions, and currencies must be able to store data in the database outline. Dynamic Calc, Dynamic Calc and Store, and Label Only are virtual members that do not store data. There is no benefit to running the copy of the currency conversion calculation script if the target version has virtual members because Planning discards the results of the calculation for these members.

To convert currencies correctly, the first time a currency conversion is launched, administrators must run the copy currency rates calculation script and the currency conversion calculation script. After launching the `HSPCrtB.csc` or `HspCrtT.csc` copy calculation script, you must launch them again if you change the database outline (for example, by adding or changing exchange rates, account rate types, versions, scenarios, accounts, or user-defined dimension members).

To create the copy currency calculation script for calculation scripts, you must regenerate currency conversion calculation scripts.

Optimizing Performance

Use these methods to optimize performance:

- Strategically assign dimensions as dense or sparse and order them from most to least dense (see [About Sparse and Dense Dimensions](#), [About Reordering Dimensions](#), and [Setting Dimension Density and Order](#)).
- Design business rules to execute within two minutes. Schedule longer-running business rules to execute in the background (see [Setting Background Processing](#)).
- Set the **Suppress Missing Blocks** option (see [Setting Form Grid Properties](#)).
- Set the Oracle Hyperion Planning property `OLAP_MAX_CONNECTIONS` to 20 if you have 100 users. If you need to after careful testing, increase the setting to 100 for 500 users (see [Setting Application and System Properties](#)).
- Optimize the JDBC connection pool settings (see [Configuring JDBC Connection Pooling](#)).
- Increase the timeout value for SQL queries (see [Optimizing SQL Relational Databases](#)).
- If you use the WebLogic Server, check its performance settings (see [Optimizing WebLogic Server Parameters](#)).

- Tune Java Virtual Machine (JVM) parameters (see [500 Error Message](#)).
- Optimize Windows network parameters (see [Optimizing Windows Network Parameters](#)).

About Reordering Dimensions

The order of dimensions is critical for the structure and performance of a Oracle Hyperion Planning application. Optimize performance when ordering dimensions:

- Make Period and Account dense, and order dense dimensions from most to least dense. The most dense is usually Period, followed by Account. Dense dimensions calculate faster than sparse dimensions.
- Separate sparse dimensions into aggregating and nonaggregating dimensions. Place aggregating dimensions before nonaggregating dimensions. Order sparse dimensions from most to least dense. Aggregating dimensions, such as Entity, consolidate children into the parent to create new data. Nonaggregating dimensions, such as Scenario, do not consolidate children to create data.

Allocating Memory for Supporting Detail Cache

To improve performance when users change planning unit status, you can specify the amount of memory for the supporting detail cache. To change the default memory allocation for supporting detail cache, increase the `SUPPORTING_DETAIL_CACHE_SIZE` property from the default of 20.

Tip:

If the Supporting Detail Detection Cache reaches 75% or greater, Oracle recommends that you decrease its size to 60%.

To view supporting detail cache usage, see [Viewing Usage Statistics](#). To change its value, see [Setting Application and System Properties](#).

Configuring the Data and Index Cache Sizes

If your system has enough memory, before you create an application, to improve performance, you may want to configure the `essbase.cfg` file to increase the index cache to 256 MB or more and increase the data cache to 2 GB or more. Changing these settings does not affect existing applications. You can increase the index and data cache for each application using either Oracle Essbase Administration Services or manually using MaxL statements. For example, using MaxL:

```
alter databaseName planType set data_cache_size 2000MB
alter databaseName planType set index_cache_size 256MB
```

Note:

Ensure that your system has enough memory for these suggested settings. If it does not, lower the settings. For more information, see *Oracle Essbase Database Administrator's Guide*.

Other Performance Optimization Tips

- Set upper-level members in a dense dimension to Dynamic Calc.
- Clear unnecessary or historical data. For example, move historical data to a separate plan type to reduce the database size in your current plan type.
- Routinely defragment the Oracle Essbase database.
- Split large forms into multiple smaller forms having fewer rows and columns. Organize related forms into folders.
- Using account annotations affects performance, so use them sparingly.
- The first time Oracle Hyperion Planning is launched and forms are opened, caches are loaded, which takes more time than in subsequent sessions. So, before your users start using Planning, launch Planning and open the most commonly used forms to reduce the loading time for your users.

Backing Up Applications and Application Databases

Back up your applications and application databases on a daily basis. Also back up before:

- Refreshing applications
- Moving applications to another server
- Upgrading applications
- Key planning milestones

Backing up applications and their related application databases consists of:

- Backing up the application in Oracle Essbase
- Backing up the relational database for Oracle Hyperion Planning and Oracle Hyperion Financial Reporting
- Backing up required components of Planning

For detailed instructions, see the *Oracle Enterprise Performance Management System Backup and Recovery Guide*.

5

Importing and Exporting Data and Metadata

Related Topics

- [Overview](#)
- [Working with the Outline Load Utility](#)
- [Working with Planning Import and Export](#)
- [Clearing EPM Oracle Instance Tmp Directory Files](#)
- [Enabling Data Load of Essbase Data](#)

Overview

Related Topics

- [About Loading Dynamic Members](#)
- [Loading Tools](#)

About Loading Dynamic Members

Child members that you load using either of the Outline Load utilities or Oracle Hyperion Enterprise Performance Management System Lifecycle Management under parent members enabled for dynamic children, are added as dynamic child members if there are dynamic member placeholders in Oracle Essbase. After the placeholders are full, any remaining children are added as normal members, and cannot be used until the database is refreshed.



Note:

If you import a parent member that is enabled for dynamic children and its child members simultaneously, during the same import, the child members are loaded as normal members. This is because the database must be refreshed to create the placeholders in Essbase. For more information about dynamic members, see [About Dynamic Members](#).

Loading Tools

Table 5-1 Metadata Load Tools

Metadata Load Tool	Comments
Outline Load utility	See Working with the Outline Load Utility .
Oracle Hyperion Planning Import and Export	See Working with Planning Import and Export .

Table 5-1 (Cont.) Metadata Load Tools

Metadata Load Tool	Comments
Oracle Hyperion Financial Data Quality Management, Enterprise Edition	Loads from supported FDMEE source systems. See the <i>Oracle Hyperion Financial Data Quality Management, Enterprise Edition Administrator's Guide</i> .

Table 5-2 Data Load Tools

Data Load Tool	Comments
Outline Load utility	Loads numeric, date, and text data values. See Working with the Outline Load Utility .
Planning Import and Export	See Working with Planning Import and Export .
FDMEE	Loads from supported FDMEE source systems. See the <i>Oracle Hyperion Financial Data Quality Management, Enterprise Edition Administrator's Guide</i> .

Working with the Outline Load Utility

The Outline Load utility can be used to import metadata and data for Account, Period, Year, Scenario, Version, Currency, Entity, user-defined dimensions, attributes, UDAs, exchange rates, Smart Lists, and planning unit hierarchies from a flat file or a relational data source. You can also export metadata and data to a flat file or export metadata to a relational data source using the Outline Load utility.

Note:

To import or export data or metadata without using a command line, see [Working with Planning Import and Export](#). Oracle Hyperion Planning Import and Export supports only imports or exports to or from a flat file.

Use these general steps to load information with the Outline Load utility:

1. When loading data, set the `DIRECT_DATA_LOAD` and `DATA_LOAD_FILE_PATH` system properties.
2. **Optional:** Create a command properties file (`.properties`) that contains command line arguments.
3. Create a load file for each dimension or set of data that you want to load.
4. Test the load files, and then run the utility.

For detailed instructions, see these topics:

- [Command Properties File](#)
- [Generating Load Files](#)
- [Importing Data and Metadata Using a Flat File](#)

- [Importing Data and Metadata from a Relational Data Source](#)
 - [Exporting Data and Metadata from a Planning Application to a Flat File](#)
 - [Command Line Parameters for the Outline Load Utility](#)
 - [Dimension Properties](#)
5. **Optional:** Clear any temporary files that were generated during the load. See [Clearing EPM Oracle Instance Tmp Directory Files](#).

Command Properties File

You can optimize command lines by storing command line arguments in a command properties file (`.properties`), then use the `/CP:` parameter in the command line to refer to that file when you execute the Outline Load utility. For example, you can use a command properties file to run the same commands on multiple applications. You would only need to change the `/A` parameter in the command line for each import.

Reasons to consider using a command properties file:

- Command lines are shorter and easier to manage
- Readability
- Ease of use
- You can bundle switches for a common application

The Java `.properties` file contains entries that are key-value pairs. The key value pairs are separated with a colon (`:`) or an equals sign (`=`); for example:

- `Key:value` OR `/I:c:/tmp/anInputCSVTextFile.txt`
- `Key=value` OR `/cp= c:/tmp/anInputCSVTextFile.txt`

Entries in the properties file are used as command switches for the utility. For arguments that appear in both the properties file and in the command line, command line arguments take precedence. The command line and properties file switches are merged at runtime and the results of the parsing are displayed in the log.

For example, the following command line:

```
/A:tr2 /U:admin /M /K /8 /DF:mm-dd-yyyy /I:c:/tmp/loadAccounts.csv /
D:Account /CP:c:/tmp/myProps.properties
```

And the `myProps.properties` file containing the following arguments:

```
/DF:yyyy-mm-dd
```

```
/D:Entity
```

```
/T
```

Would output these log messages showing the result of the parsing:

- Properties file arguments: `/DF:yyyy-mm-dd /D:Entity /T`

- **Command line arguments:** /A:tr2 /U:admin /M /K /8 /DF:mm-dd-yyyy /I:c:/tmp/loadAccounts.csv /D:Account /CP:c:/tmp/myProps.properties
- **Submitted (merged) command line:** /A:tr2 /U:admin /M /K /T /8 */DF:mm-dd-yyyy /I:c:/tmp/loadAccounts.csv */D:Account /CP:c:/tmp/myProps.properties

 **Note:**

An asterisk denotes that the command switch was found in both the properties file and in the command line and that the command line version has superseded the command property file version.

Generating Load Files

Generating a Data Load File

When loading data with the Outline Load utility, you can specify driver members in the load file and run the utility with the /TR option.

 **Note:**

You can also load to driver members that are specified on the Oracle Hyperion Planning Data Load Administration page.

Generate a comma-separated or tab-separated data load file containing these columns:

- **Driver Member:** The member into which data is loaded. You can have one driver dimension per load. Multiple members can be defined for the driver dimension. The value is passed as a string representing a numeric value, or, if a Smart List is bound to the member, as a Smart List value.
- **Point-of-View:** All other dimensions required to determine the intersection for which to load the data. (If you are using /TR, include all of the members except the driver member.) The data load automatically performs cross-product record creations based on the dimension parameters in the point of view (POV). The load file creates and loads the data record for each relevant cell intersection. The value is passed as a string. The POV accepts a comma-separated list of members, including member functions. For example, children(Q1) is expanded to Jan, Feb, Mar during the load. The corresponding records are generated based on the cross product of all member combinations and the data value.
- **Data Load Cube Name:** The name of the plan type to which data is being loaded. The value is passed as a string. Values include any plan types specified in the application, such as Plan1.

Example 1: In this example, Account was selected as the Data Load dimension on the application's **Data Load Administration** page. Period was selected as the Driver Dimension, and Jan was selected as the Driver member.

```
Account,Jan,Point-of-View,Data Load Cube Name
```

```
"acct1,12,Local,ent1,Current,Ver1,FY08",Plan1
```

Example 2: In this example, Entity was selected as the Data Load dimension on the application's **Data Load Administration** page. Account was selected as the Driver Dimension, and Account members aUnspec, aSmart, aDate, and aText were selected as the driver members. This load file loads data into the intersection of e1, the point of view, and the Account driver members, aUnspec, aSmart, aDate, and aText.

```
Entity,Operation,Data Load Cube Name,aUnspec,aSmart,aDate,aText,Point-of-View
```

```
e1, ,Plan1,77,smart1,12-22-2008,textValue,"USD,Jan,Current, BUVersion_1, FY07"
```

Assuming these values for the driver members:

- **aUnspec:** Data Type Unspecified (numeric), value 77
- **aSmart:** Data Type Smartlist,value smartlist entry 'smart1'
- **aDate:** Data Type Date,value 12-22-2008
- **aText:** Data Type Text,value 'textValue'

If **DIRECT_DATA_LOAD** is set to False, the example would generate this data load file:

```

Currency Version Scenario Year Entity Period Account HSP_Rates

77 USD BUVersion_1 Current FY07 e1 Jan aUnspec HSP_InputValue

1 USD BUVersion_1 Current FY07 e1 Jan aText HSP_InputValue

20081222 USD BUVersion_1 Current FY07 e1 Jan Date HSP_InputValue

1 USD BUVersion_1 Current FY07 e1 Jan aSmart HSP_InputValue

```

Example 3: Specify drivers directly in the load file.

```
Value,Driver Member,Point-of-View,Data Load Cube Name
```

```
14,a1,"Jan,Local,e1,Current,Version1,FY08",Plan1
```

```
s11_value2,a2,"Jan,Local,e1,Current,Version1,FY08",Plan1
```

```
OutlineLoad /A:acpt1 /U:admin /M /I:c:\outline\data.csv /TR /L:c:/OutlineLogs/  
outlineLoad.log /X:c:/OutlineLogs/outlineLoad.exc
```

To add comments to a load file, see [Comments in Load Files](#).

Generating a Metadata Load File

When loading metadata, the load file must contain a header record that lists the dimension, such as Account, and the member properties used by subsequent metadata records. For example, for Account, you can specify which account to load, a default alias, the operation to perform, and so on. Header records are case sensitive. They can appear in any order.

The next lines in the load file contain metadata records, listed in the order designated by the header record. Each metadata record contains a comma- or tab-separated list of property values that correspond to the header record entries. For detailed information on the properties available for each Oracle Hyperion Planning member, see [Dimension Properties](#).

Example: This load file loads an Entity dimension with the required header record and three data records. The header record specifies the member to be loaded (Entity), the parent member (Parent) into which to load the member, and the Data Storage property to assign to the member.

```
Entity, Parent, Data Storage
```

```
e1, Entity,
```

```
e2, ,
```

```
e1, e2, Shared
```

Using this load file would result in this outline, assuming that no other members exist:

```
Entity
```

```
  e1
```

```
  e2
```

```
    e1 (Shared)
```

The first data record (e1, Entity) loads Entity member e1 as a child under the root member Entity. Unspecified values assume the default. For example, if data storage is not specified, it assumes the default value, Never Share. The next data record (e2,) loads Entity member e2 under the dimension root member because no parent is specified, and sets data storage to Never Share. The last data record (e1, e2, Shared) loads a shared member of e1 under member e2, and sets data storage to Shared.

When loading the Currency dimension with the Default Currency Symbol, if a currency is added without a symbol specified, the symbol is set to that of a pre-defined currency of the same name (or, if the name does not match a pre-defined currency, to the currency code of the currency being added).

To add comments to a load file, see [Comments in Load Files](#).

Comments in Load Files

Comments are supported in Outline Load utility input CSV files.

- For single line comments, place the hash character as the first character on the line; for example, # comment.
- Blank lines are ignored.
- Block comments are delineated by a start comment block indicator:#!– and terminated on a separate line with an end block indicator: #--! Intervening lines need not be commented.

For example:

```
#!–start of comment block
```

```
Comment within block
```

```
Another comment within block
```

```
#--! End of comment block
```

**Note:**

Comment blocks cannot be nested. In addition, `#!-- HEADERBLOCK` is a reserved, Oracle Hyperion Planning internal, comment block header.

Load File Considerations

Consider these points when working with load files:

- Users must write macros to handle new line characters in Excel. Users cannot open and modify Outline Load utility export files that contain multiline formulas in Excel.
- If member names contain commas, you must use a tab-delimited file when performing a data export.
- If a member has the same name as one of the member properties (for example, Description), Outline Load utility data and metadata export may have unexpected results.
- If saving or modifying metadata or data comma-separated values (CSV) files generated by the Outline Load utility export/import in Microsoft Excel, be aware that Excel does not properly handle some formatting features in the CSV file. For example, Outline Load utility places the POV member information in double quotation marks and treats it as one column, while Excel treats the POV members as separate columns. Saving the file in Excel will place an additional set of quotation marks around the POV members and will add commas to the top header row. Outline Load utility will not recognize this as a valid format when the file is imported back. Oracle recommends editing and saving CSV export files in Notepad, Wordpad, or another text editor.
- For each dimension in the application, you create a load file with fields corresponding to the dimension properties. Each load file can contain members for only one dimension. You can define multiple members for each dimension.
- The required fields are different for each dimension being loaded. See [Dimension Properties](#).
- The member name must be included as a field in the load file. If the member property value is not specified, the application default value for the property is used.
- When adding new members, unspecified values assume a default value or are inherited from the parent member's property value as appropriate. If the member exists and no value is specified, it is left as is.
- When you load a member that already exists in the Oracle Hyperion Planning application (for example, to change a property) and a parent is not specified in the load file, the member is left under the existing parent. If a new parent is specified, the member is moved under the new parent.
- The planning unit hierarchy load behavior differs from other dimension loads in that the import file specifies a complete replacement of the hierarchy instead of incremental changes, as is the case when loading other dimensions. Loading planning unit hierarchies first deletes all members of the planning unit hierarchy, and then adds each member specified in the input file as a new member. It is important to keep in mind that a planning unit hierarchy load deletes an existing member and its children from the hierarchy if the member is not specified in the input file.
- To specify a null value, you can use the reserved value, `<none>`, for example, to delete an attribute assignment.
- Header record fields can appear in any order.

- Only one dimension can be loaded per load file.
- Column headers in the load file are case-sensitive.
- The records are loaded one by one. If a record fails to load, its associated exception is written to the exception file and the load process resumes with the next record.
- If errors are logged when loading a year, and the year was loaded into the application, its properties may not be what was specified for it in the load file. Correct the load file record and reload the year to set its properties correctly.
- Parent members must exist or be loaded before their child members. In most cases, the load file must be sorted in parent-child order, either explicitly or by using /H.
- Data values containing commas and quotation marks must be enclosed in quotation marks. These examples show how commas and quotation marks are interpreted.

Table 5-3 Examples of Data Values Containing Commas and Quotation Marks

Value	Interpretation
"quote""quote"	quote"quote
""quotedstring""	"quotedstring"
""",quoted,"",string,""	",quoted," ,string,"
""",quoted,"""",string,"""	",quoted,"" ,string,""

- Member names with parenthesis will be treated as functions.

Importing Data and Metadata Using a Flat File

Loading Data

When loading data with the Outline Load utility, there are two ways to specify driver members. You can load to driver members that are specified on the Oracle Hyperion Planning Data Load Administration page, or you can specify driver members in the load file and run the utility with the /TR option.

If you load data with the /TR option, the load file must list the driver member and all other members under the Point-of-View column, regardless of their location on the form. For example, if Jan or Descendants (YearTotal) are columns in a form, they must be specified in the Point-of-View column. When using /TR, you can load one value per row in the load file. You can include multiple rows, but you can specify only one data value per row. See [Generating Load Files](#).

Caution:

Following these steps can affect data in the database. The Planning `DIRECT_DATA_LOAD` system property enables data to be loaded directly to Oracle Essbase. In the current release, this property is set to true by default, and data is loaded directly into Essbase. To prevent data from being loaded directly into Essbase, set `DIRECT_DATA_LOAD` to false.

To load data with the Outline Load utility:

1. Back up the application and application databases before loading information. See the *Oracle Enterprise Performance Management System Backup and Recovery Guide*.
2. Set Planning System properties.
 - a. Log in to the Planning application.
 - b. Select **Administration**, then **Application**, then **Properties**, then click the **System Properties** tab.
 - c. Set the `DIRECT_DATA_LOAD` and `DATA_LOAD_FILE_PATH` properties:
 - If `DIRECT_DATA_LOAD` is set to `True`, or if you do not specify a value for this property, information is loaded directly into Essbase while the load file records are processed. For this method to work correctly, the outlines maintained in Planning and Essbase must be synchronized. The load file must not specify any Planning outline changes unless they have already been refreshed to Essbase.
 - If `DIRECT_DATA_LOAD` is set to `False`, the Outline Load utility processes the load file that you created to generate a data file (.TXT) and rule file (.RUL). This way, the Planning and Essbase outlines do not need to be synchronized because data is not loaded at this time. You can refresh the changes at a convenient time to propagate the metadata changes to Essbase, and then load data directly into Essbase (for example, using Oracle Essbase Administration Services).

In most cases, set `DIRECT_DATA_LOAD` to `False`, and set `DATA_LOAD_PATH` to the location and name that will be used for the generated data and rules files, for example, `C:/myDirectory/App1.txt`. Ensure that these properties are set in the **System Properties** tab in the **Manage Properties** page.
 - d. Restart the Planning application server.
3. If you want to load to driver members that are specified in Planning, set the driver members as described in this step. Otherwise, skip to the next step.
 - a. Log on to the Planning application for which data will be loaded.
 - b. Select **Administration**, and then **Data Load Settings**.
 - c. Select a dimension from the **Data Load Dimension** list (such as **Account**). This is the dimension for which you want to load data. For example, it may appear as a row in a Planning form.
 - d. Select a dimension from the **Driver Dimension** list (such as **Period**).
 - e. Click the member selection icon to select members of the **Driver Dimension** (such as **Jan, Feb, March**). For example, these members may appear as columns in a Planning form.
4. Generate a comma-separated load file. See [Generating a Data Load File](#).
5. Test the load file and run the utility.

- a. Locate the utility, installed in the `planning1` directory.

For the full path to `planning1`, see [About EPM Oracle Instance](#).

- b. To confirm that the load file parses without any errors, run the utility using the `/N` parameter, and check the outline log file to be sure no error messages were generated. Running the utility with `/N` does not load data or metadata, but ensures that the load file parses successfully. For example, you could use this command line to check the load file for a Planning application called `test`:

```
C:\EPM_ORACLE_INSTANCE\Planning\planning1>OutlineLoad /A:test /
U:admin /M /N /I:c:\outline\data3.csv /D:Entity /L:c:/outlineLoad.log /
X:c:/outlineLoad.exc
```

- c. You can then run the utility from the command prompt without /N, using the case-sensitive command, one space, and the appropriate parameters. For example:

```
C:\EPM_ORACLE_INSTANCE\Planning\planning1>OutlineLoad /A:test /
U:admin /M /I:c:\outline\data3.csv /D:Entity /L:c:/outlineLoad.log /X:c:/
outlineLoad.exc
```

If you are loading data without specifying driver members within Planning, you can run the utility including /TR. For example:

```
C:\EPM_ORACLE_INSTANCE\Planning\planning1>OutlineLoad /A:test /
U:admin /M /N /I:c:\outline\data3.csv /TR /D:Entity /L:c:/
outlineLoad.log /X:c:/outlineLoad.exc
```

For detailed information on the parameters available for use with the Outline Load utility, see [Dimension Properties](#).

Loading Metadata

Metadata for applications using Oracle Hyperion Planning application administration can be loaded for Account, Period, Year, Scenario, Version, Currency, Entity, user-defined dimensions, attributes, UDAs, and Smart Lists. Values can also be loaded for exchange rates. However, because exchange rate values are loaded into the Planning relational tables, not directly into Oracle Essbase, the procedure for loading metadata still applies.

The utility loads one record at a time. If a record fails to load, a message is written to the exception file, and the load process resumes with the next record. When new members are added, unspecified properties assume the default value or inherit the parent member's property as appropriate. If a member exists and no property value is specified, the property is left as is.

To load metadata:

1. Generate the load file. See [Generating a Metadata Load File](#).
2. Test the load file and run the utility.
 - a. Locate the Outline Load utility, installed in the `planning1` directory.
For the full path to the `planning1` directory, see [About EPM Oracle Instance](#).
 - b. To confirm that the load file parses without any errors, run the utility using /N, and check the log file to be sure no error messages were generated. For example:


```
C:\EPM_ORACLE_INSTANCE\Planning\planning1>OutlineLoad /A:test /
U:admin /M /N /I:c:\outline\ent.csv /D:Entity /L:c:/outlineLoad.log /
X:c:/outlineLoad.exc
```
 - c. You can then run the utility from the command prompt, using the case-sensitive command, one space, and the appropriate parameters.

For detailed information on the parameters available for the Outline Load utility, see [Dimension Properties](#).

Importing Data and Metadata from a Relational Data Source

You can also import metadata and data from a relational database source. The import of metadata and data contained in a relational database is achieved by supplying a query on the external database that returns a result set equivalent to the flat file input format. A query and database connection information needs to be provided.

Oracle recommends that users employing this functionality are familiar with relational databases, the SQL query language, and JDBC connection properties. For Oracle Hyperion

Planning applications, examples of these properties can be found in the `HSPSYS_DATASOURCE` table in the system database, which by default is located in the relational database that you create when selecting the Configure Database task under Planning in EPM System Configurator.

These command line parameters are available for users using the Outline Load utility to import from a relational data source:

- `/IR[:RDBConnectionPropertiesFileName]`
- `/IRA`
- `/RIQ:inputQueryOrKey`
- `/RIC:catalog`
- `/RID:driver`
- `/RIR:url`
- `/RIU:userName`
- `/RIP:password`



Note:

The `/IR` and `/RIQ` parameters are exclusive.

For descriptions of these parameters, see [Command Line Parameters for the Outline Load Utility](#).



Caution:

Before performing an import operation, ensure you back up the Planning relational store and the Oracle Essbase data. See the *Oracle Enterprise Performance Management System Backup and Recovery Guide*.

Importing Metadata

The following is an example command line for running the Outline Load utility using the `myprop_relational.properties` file (detailed below) as input:

```
OutlineLoad /CP:c:/myprop_relational.properties
```

The file will import Account members from the `PS2ORAU` application shown in the JDBC connection parameters into an application named `Test_300`. The properties file can contain several relational queries. The `/RIQ` parameter determines which query will be executed by the Outline Load utility.

For a description of `/RIQ`, see [Command Line Parameters for the Outline Load Utility](#).

Example 5-1 Example: The myprop_relational.properties File

```

/A:Test_300
/U:admin
/RIQ: ACCOUNT_QUERY

/D:Account
#
ACCOUNT_QUERY=SELECT O.OBJECT_NAME as Account, (select object_name from
hsp_object where object_id = O.PARENT_ID) as Parent from HSP_ACCOUNT A,
HSP_MEMBER M, HSP_OBJECT O LEFT OUTER JOIN HSP_STRINGS S ON O.DESCRPTION =
S.STRING_SEQ WHERE O.OBJECT_ID=M.MEMBER_ID AND M.MEMBER_ID = A.ACCOUNT_ID
AND M.MEMBER_ID <> M.DIM_ID ORDER BY O.POSITION
#
ENTITY_QUERY=SELECT O.OBJECT_NAME as Entity, (select object_name from
hsp_object where object_id = O.PARENT_ID) as Parent from HSP_ENTITY E,
HSP_MEMBER M, HSP_OBJECT O LEFT OUTER JOIN HSP_STRINGS S ON O.DESCRPTION =
S.STRING_SEQ WHERE O.OBJECT_ID=M.MEMBER_ID AND M.MEMBER_ID = E.ENTITY_ID AND
M.MEMBER_ID <> M.DIM_ID ORDER BY O.POSITION

## jdbc connection
/RIC:PS2ORAU
/RIR:jdbc:oracle:thin:[scl34390]:1521:orcl
/RID:oracle.jdbc.OracleDriver
/RIU:PS2ORAU
/RIP:password

#end myprop_relational.properties file

```

In the following example, an attribute dimension in both source and target applications has a name with more than 30 characters, Size012345678901234567890123456789, which exceeds the Oracle column header limit. The query in the following example is used to create a result set from the source to be imported into the target application. The long dimension name must be aliased to column O.OBJECT_NAME to allow OLU to use this column as the attribute dimension name.

Example 5-2 Example: Properties File for Importing from a Relational Data Source to a Planning Application

```

/A:TARGET
/U:admin
/IR

/DA:Size012345678901234567890123456789:Entity

/C2A: (OBJECT_NAME,Size012345678901234567890123456789)

ATTRIB_DIM_VAL_QUERY_ORACLE1=SELECT O.OBJECT_NAME, (select object_name from
PS3ORA.hsp_object where object_id = O.PARENT_ID) as Parent from
PS3ORA.HSP_OBJECT O WHERE O.OBJECT_ID IN (select AM.ATTR_MEM_ID from
PS3ORA.hsp_attribute_member AM, PS3ORA.HSP_OBJECT O where AM.ATTR_ID =(select
OBJECT_ID from PS3ORA.hsp_object where
OBJECT_NAME='Size012345678901234567890123456789')) ORDER BY O.POSITION

```

Example 5-3 Example: Properties File for Importing with Alias from a Relational Data Source to a Planning Application

```

/A:expe
/U:admin
/IRA
/D:Entity
/C2A:(Member as
Entity,anotherReallyLongAliasForAColumnNameItsForEntityThisTimeDifferentFromExport), (Parent, anotherAliasForParent)

/RIQ:select
anotherReallyLongAliasForAColumnNameItsForEntityThisTimeDifferentFromExport,
Parent from test

```

Assume the following with regards to the previous example:

- The 'Test' table has two columns, Member and Parent, and it has one entry:

Member	Parent
Ex	Entity

- The Entity dimension has one member, e1, under the root (Entity).

After running the previous query, the member Ex is added to or updated in the Entity dimension.

On import, the aliases for columns are handled as a straight string replacement which is why the "as" clause can work in this case. If the Member column was named Entity, you would drop the 'Member as.'

Example 5-4 Example: Using the /C2A Parameter for Importing Aliases into an Alias Table

```

/A:TARGET
/U:admin
/IR
/D:Entity

/RIQ:ENT_ALIAS_ASSIGNMENT_QUERY

/C2A:(AliasTableName, Alias: LongAliasTableName012345678901234567890123)

ENT_ALIAS_ASSIGNMENT_QUERY=SELECT O.OBJECT_NAME as Entity, (select
object_name from hsp_object where OBJECT_ID=O.PARENT_ID) as Parent, (select
OBJECT_NAME from HSP_OBJECT where OBJECT_ID = AL.ALIAS_ID) as AliasTableName
from hsp_object O, HSP_ALIAS AL where (AL.ALIAS_TBL_ID = (select OBJECT_ID
from HSP_OBJECT where OBJECT_NAME = '
LongAliasTableName012345678901234567890123') AND O.OBJECT_ID=AL.MEMBER_ID)

```



Note:

In the previous example, both the target Oracle Hyperion Planning application and the source contain a long alias table name.

Example 5-5 Example: Using the /c2A Parameter to Map Column Names to Property Names

```

/RIQ:select
c1,c2,c3,c4,c5,c6,c7,c8,c9,c10,c11,c12,c13,c14,c15,c16,c17,c18,c19,c20,c21,c22
,c23,c24,c25,c26,c27 from table28

/c2a:(c1, Account), \
(c2, Parent), \
(c3, Alias: Default), \
(c4, Valid For Consolidations), \
(c5, Data Storage), \
(c6, Two Pass Calculation), \
(c7, Description), \
(c8, Formula), \
(c9, UDA), \
(c10, Smart List), \
(c11, Data Type), \
(c12, Hierarchy Type), \
(c13, Enable for Dynamic Children), \
(c14, Number of Possible Dynamic Children), \
(c15, Access Granted to Member Creator), \
(c16, Account Type), \
(c17, Time Balance), \
(c18, Skip Value), \
(c19, Exchange Rate Type), \
(c20, Variance Reporting), \
(c21, Source Plan Type), \
(c22, Plan Type (Plan1)), \
(c23, Aggregation (Plan1)), \
(c24, Plan Type (Plan2)), \
(c25, Aggregation (Plan2)), \
(c26, Plan Type (Plan3)), \
(c27, Aggregation (Plan3))

```

Importing Data

Note the following important information when performing data import operations:

- The data types of the driver members must be the same as the imported value data types or data errors will occur. Date fields must specify the format with a /DF switch and the data type of the driver member must be type date. Similarly, a Smartlist member must be of type smartlist if the Smartlist value exists in the application and in the specified Smartlist. Text values must be bound to driver members of type text.
- In addition to driver member types, the evaluation order of dimensions must be set so that the driver member values are evaluated correctly.
- Outline Load utility data import does not import #missing values. To ensure the correct Oracle Essbase cells will contain #missing values, clear the block (which sets all cells in the block to #missing) to be loaded with the /ICB command on all data imports. See [Clearing an Essbase Block Using the /ICB Parameter](#).

To import data from a relational database:

1. Back up the Oracle Hyperion Planning relational store and the Essbase data. See the *Oracle Enterprise Performance Management System Backup and Recovery Guide*.

2. Create a relational table in an application.

The following graphic is an example of a relational table named `Data_Table1` located in database `Test_300A`:

Entity	Data Load Cub...	Point-of-View	aCur	aNonCur	aPer	aSL	aDate	aText	aUnspec
31	Plan1	"FY12, Ron,Current, BU Version_1, Local,Jan"	70	80	.75	b	6-15-2013	text3	10
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

3. Create a Planning application form.

The following graphic is an example of a Planning form named `Test_300`:

4. Create a `.properties` file.

 **Note:**

Effective column heading names are obtained from the column name on the select statement, which can be overridden with the "as" clause.

The following is an example of a properties file named `myprop_relational.properties`:

```

/A:TEST_300
/U:admin
/IR
/RIQ:DATA_QUERY2
/D:Entity

DATA_IMPORT_QUERY3 = SELECT Entity ,"DATA LOAD CUBE NAME" as 'Data Load
Cube Name',"AUNSPEC" as "aUnspec","ACUR" as "Salary_aCur","ANONCUR" as
"aNonCur","APER" as "aPer","ASL" as "aSL","ADATE" as "aDate","ATEXT" as
"aText","POINT-OF-VIEW" as "Point-of-View" from Data_Table1

/RIC:Test_300A
/RIR:jdbc:weblogic:sqlserver://[scl34390]:1433;DatabaseName=Test_300A
/RID:weblogic.jdbc.sqlserver.SQLServerDriver
/RIU:sa
/RIP:password

```

5. Run the Outline Load utility with the `.properties` file created in the previous step.

For example, running the following command line will import data from the relational table Data_Table1 located in database Test_300A into the Planning form in Planning application Test_300:

```
OutlineLoad /CP:c:/myprop_relational.properties
```

The resulting Planning application form:

Salary_aCur	aUnspec	aNonCur	aPer	aSL	aDate	aText
70	10	80	75%	b_Label	6/15/13	text3

6. Check Essbase log files and validate the results of the import operation.

Clearing an Essbase Block Using the /ICB Parameter

Outline Load utility data import does not import #missing values. To ensure the correct Oracle Essbase cells will contain #missing values, clear the block (which sets all cells in the block to #missing) to be loaded with the /ICB command on all data imports.

For example, the value for ACUR in the following graphic is blank or #missing:

AUNSPEC	ACUR	ANONCUR	APER	ASL	ADATE	ATEXT	ENTITY	POINT-OF-VIEW	DATA LOAD CUBE NAME
30		50	0%	07-15-2012	text1	e1	FY12,Current, B...	Plan1	

And the value for aCur in the resulting Oracle Hyperion Planning form will be blank or #missing:

aUnspec	aCur	aNonCur	aPer	aSL	aDate	aText
30		50	0%	b_Label	7/15/12	text1

The following is an example of a properties file named my_prop.properties which includes the /ICB parameter:

Command line: OutlineLoad /CP:c:/my_prop.properties

Properties file:

```
/A:DB2APP
/U:admin
```

```
/ICB:e1,"aUnspec,aCur,aNonCur,aPer,aSL,aDate,aText","FY12,Current, BU
Version_1, Local,Jan",Plan1
```

```
#Use /IRA switch if relational table from which data is imported is in the
same relational database as the Planning app (e.g. DB2APP)
```

```
/IRA
/RIQ:DATA_IMPORT_QUERY3
```

```

/D:Entity
#/C
#/M
DATA_IMPORT_QUERY3 = SELECT "ENTITY" as "Entity" ,"DATA LOAD CUBE NAME" as
"Data Load Cube Name","AUNSPEC" as "aUnspec","ACUR" as "aCur","ANONCUR" as
"aNonCur","APER" as "aPer","ASL" as "aSL","ADATE" as "aDate","ATEXT" as
"aText","POINT-OF-VIEW" as "Point-of-View" from DataTable1

```

Setting Outline Load Utility Drivers Using the /SDM Parameter

Before importing data, you can use the /SDM parameter to set Load Dimension and Driver Members through Outline Load utility. You can do this instead of using the Administration/Data Load Settings user interface in the Oracle Hyperion Planning application.

The following is an example of a properties file named `my_prop.properties` which includes the /SDM parameter:

Command line: `OutlineLoad /CP:c:/my_prop.properties`

Properties file:

```

/A:Test_300
/U:admin

/ICB:"aUnspec,aPer","Jan, Feb, Mar","FY12, Current, BU Version_1,
Local,e1,Ron",Plan1

#Use /IRA switch if relational table from which data is imported is in the
same relational database as the Planning app (e.g. Test_300)
/IRA

/SDM:Account,"Jan, Feb, Mar",Plan1
/RIQ:DATA_IMPORT_QUERY5

/D:Account

DATA_IMPORT_QUERY5 = SELECT Account ,"Data Load Cube Name",
Jan, Feb, Mar,"Point-of-View" from Data_Table2

```

The following example is a relational table:

SQLQuery3.sql ...300 (sa (60))*		SCL34390.Test...bo.Data_Table2				
	Account	Data Load Cub...	Point-of-View	Jan	Feb	Mar
	aUnspec	Plan1	"FY12, Ron,Curr...	10	20	30
	aPer	Plan1	"FY12, Ron,Curr...		0.65	0.85
	NULL	NULL	NULL	NULL	NULL	NULL

The resulting Planning application form:

	Jan	Feb	Mar
aUnspec	10	20	30
aPer		65%	85%

Exporting Data and Metadata from a Planning Application to a Flat File

Exporting Data with the Outline Load Utility

Running the Outline Load utility with the `myprop_dataexport.properties` file exports data contained in the intersection specified by the `/EDD` parameter in the `Test_300` Oracle Hyperion Planning application into a flat file called `output_file.1-1.csv` on the `c:\` drive as specified by the `/ED` parameter.

The `myprop_dataexport.properties` file:

```
/A:Test_300
/U:admin

/ED:c:/output_file
/EDD:aUnspec, "Jan, Feb, Mar", "FY12, Current, Local, e1, BU Version_1, Ron", Plan1
```

For descriptions of the `/ED` and `/EDD` parameters, see [Command Line Parameters for the Outline Load Utility](#).

If the `Test_300` application contains the following form, running the Outline Load utility with the `myprop_dataexport.properties` file will export data into files of the form `output_file.1-1.csv` through `output_file.n-n.csv` where `n` is the number of files generated.

	Jan	Feb	Mar
aUnspec	10	250	300

The following is an example command line for running the Outline Load utility using the `myprop_dataexport.properties` file as input:

```
OutlineLoad /CP:c:/myprop_dataexport.properties
```

The `output_file.1-1.csv` file:

```
Account, Jan, Feb, Mar, Point-of-View, Data Load Cube Name
aUnspec,10, 250, 300, "FY12, Current, BU Version_1, Local, e1, Ron", Plan1
```

Member functions can be used with the `/EDD` parameter.

If the load dimension is Employee and the driver dimension is Account, the following statement can be used to export data:

```
/EDD:Ilvl0Descendants(John),
Ilvl0Descendants(Accounts),"FY12,Current,Local,e1,BU Version_1,Jan",Plan1
```

Considerations for exporting data:

- If member names contain commas, you must use a tab-delimited file when performing a data export.
- If a member has the same name as one of the member properties (for example, Description). Outline Load utility data export may have unexpected results.
- Ensure you set up the evaluation order so the text or other type of data is displayed on the form the same way you would like to view it in the export file.



Note:

Exporting data requires the column dimension to be dense.

Exporting Metadata with the Outline Load Utility

You can use the Outline Load utility to export metadata for a dimension to a flat file for Account, Period, Year, Scenario, Version, Currency, Entity, user-defined dimensions, attributes, UDAs, exchange rates, Smart Lists, and planning unit hierarchies. For information about the parameters used with this utility, see [Command Line Parameters for the Outline Load Utility](#).

To export metadata with the Outline Load utility, specify the dimension name and provide the name of the file to which you are exporting. For example, you could use the following command to export metadata for the Entity dimension to a file called `ent_export.csv`:

```
OutlineLoad /A:Acpt1 /U:admin /-M /E:c:/ent_export.csv /D:Entity /L:c:/
outlineLoad.log /X:c:/outlineLoad.exc
```

Example: Exporting user-defined dimensions with a space in the name

When exporting user-defined dimensions with a space in the name, enclose the name in quotation marks (" "). For example, you could use the following command to export metadata for the user-defined Line Item dimension:

```
OutlineLoad /A:plnldv /U:admin /-M /E:c:/LineItemUserDim_export.csv /D:"Line
Item" /L:c:/outlineLoad.log /X:c:/outlineLoad.exc
```

Example: Exporting attribute dimensions

When exporting attribute dimensions, use the `/D` switch with the name of the attribute dimension without specifying attribute dimension type, as shown in the following example:

```
OutlineLoad /A:plnldv /U:admin /E:c:/AttribTextColor_export.csv /D:Color /
L:c:/outlineLoad.log /X:c:/outlineLoad.exc
```

Example: Exporting with new line characters in the Description text field.

New line characters are supported in text fields for Description and Formula. A field that contains new line characters must be enclosed by quotation marks, as shown in the following example:

```
Currency, Description, Data Storage
USD,          "Description Line 1
              Description Line 2
              Description Line 3", Store
EUR,          "description for EUR", Store
```

Considerations for exporting metadata:

- UDAs cannot be exported as a separate dimension. They are exported with the members to which they are assigned, similar to formulas.
- The DTS Period type is recognized, but is not supported by the Outline Load utility. If you try to export DTS information, the utility displays an error in the log file.
- If exported members have aliases in an alias table other than Default and these members are imported into another application, the alias table must be created manually in the application before the members are imported.
- If a member has the same name as one of the member properties (for example, Description). Outline Load utility metadata export may have unexpected results.

Exporting Metadata from a Planning Application to a Relational Data Source

You can use the Outline Load utility to export metadata to a relational data source.



Note:

You cannot export data to a relational data source using the Outline Load utility.

Oracle recommends that users employing this functionality are familiar with relational databases, the SQL query language, and JDBC connection properties. For Oracle Hyperion Planning applications, examples of these properties can be found in the `HSPSYS_DATASOURCE` table in the system database, which by default is located in the relational database that you create when selecting the Configure Database task under Planning in EPM System Configurator.

These command line parameters are available for users using the Outline Load utility to export to a relational data source:

- `/ER[:RDBConnectionPropertiesFileName]`
- `/ERA`
- `/REQ:exportQueryOrKey`
- `/REC:catalog`
- `/RED:driver`

- /RER:url
- /REU:userName
- /REP:password

 **Note:**

The /ER and /REQ parameters are exclusive.

For descriptions of these parameters, see [Command Line Parameters for the Outline Load Utility](#).

The export query takes the form: `INSERT INTO<tableName>[(column1, column2, ...)] VALUES (property1, property2,...)`

Notes:

- <tableName> is the name of the table into which the exported values will be inserted
- (column1, column2, ...) is an optional list of column names and the values (properties) will be inserted into the table
- (property1, property2, ...) are not strictly values, but member property names (as in the column headers)

 **Note:**

Unlike header records, the member name is specified with 'Member', not the dimension name.

For example:

Assume Planning has two members in its Entity dimension 'e1' with description 'e1's description' and 'e11' with description 'e11's description.' E1 resides under entity, e11 resides under e1. So the planning dimension looks like:

Entity

```
e1
  e11
```

Assume the destination table `my_table` has three columns: `column1`, `column2`, `column3`

Insert into `my_table` values (Member, Parent, Description) results in `my_table` having the following rows added:

```
E1      e1's description      Entity
E11     e2's description     e1
```

Which is the same as:

```
Insert into my_table (column1, column2, column3) values (Member, Parent,
Description)
```



Note:

The table must contain as many or more columns as properties specified.

There are two special operators that can be specified in the query: `<columns>` and `<properties>`. `<columns>` expands to all of the column names in the table. `<properties>` expands to all of the properties of the exported dimension, as shown with the `/M` switch.

```
Insert into my_table values (<properties>)
```

Which loads all entity properties into `my_table`.

Which is the same as:

```
Insert into my_table (column1, column2, .... column23) values (<properties>)
```

If `my_table` contained exactly the number of columns as properties, the following query would produce the same result:

```
Insert into my_table (<columns>) values (<properties>)
```

You could also use the names of table columns to extract the member properties. For example, `sample_table`, with columns: `Member`, `Parent`, and `Description`. The following queries would produce the same result as the first example:

```
Insert into sample_table values (<columns>)
```

```
Insert into sample_table (<columns>) values (<columns>)
```

Example 5-6 Example: Properties File for Exporting Metadata to a Relational Data Source

```
/A:App1
/U:admin
```

```
/ERA
```

```
/D:Entity
```

```
/REQ:ENTITY_EXPORT_QUERY1
```

```
ENTITY_EXPORT_QUERY1=INSERT INTO DataTable_Entity1 VALUES (Member, Parent,
Description, Color)
```

```
#ENTITY_EXPORT_QUERY1=Insert into PS3ORA.DataTable_Entity1 (Member, Parent,
Description, Color) values (Member, Parent, Description, Color)
```

Example 5-7 Example: Properties File for Exporting with an Alias from an Outline to a Relational Data Source

```
/A:expe
/U:admin
/ERA
/D:Entity
/REQ:ENTITY_EXPORT_QUERY1
/C2A:AliasLabel
AliasLabel:(AliasTableName, Alias: reallyLongAliasTableName30Char)

ENTITY_EXPORT_QUERY1=Insert into Test(Member, Parent, AliasTableName) values
(Member,
Parent, Alias: reallyLongAliasTableName30Char)
```

Assume the following with regards to the previous example:

- The 'Test' table has three columns, Member, Parent, AliasTableName and the table is empty.
- The Entity dimension has one member, e1, under the root (Entity).

After running the previous query, the 'Test' table will have one entry:

```
Member      Parent      AliasTableName
-----      -
e1          e1          reallyLongAliasTableName30Char
```

Note:

The `/C2A` and/or `Label` switches may be used to work around the thirty character column name limit restriction in Oracle relational tables; for example, `Alias: reallyLongAliasTableName30Char` is longer than thirty characters and not allowed as a column name in Oracle relational tables.

Command Line Parameters for the Outline Load Utility

The following command line parameters are available for the Outline Load utility. After running the utility, you can verify the results by reviewing the exception file and log file. If no errors are reported in the log file, you can then access the imported metadata and data in the application. It is not necessary to restart the application server.

```
HspOutlineLoad [-f:passwordFile] /A:application /U:userName [/
CP:commandPropertyFileName] [/M] [ [/I:inputFileName|/
IR[:RDBConnectionPropertiesFileName]||IRA|/E:outputFileName|/
ED:outputFileNameStem] [/EDD:dataExportSpecification] [/ICB:blockSpecification]
[/SDM:driverMemberSpecification] /D[U]:loadDimensionName|/
DA:attributeDimensionName:baseDimensionName|TR] [/N] [[/R] [/DPU]] [/C] [/F] [/K]
[/8] [/DF:datePattern] [/RIQ:inputQueryOrKey] [/RIC:catalog /RID:driver /
RIR:url /RIU:userName [/RIP:password]] [/X:exceptionFileName] [L:logFileName]
[/?]
```

Parameter	Description
<code>[-f:passwordFile]</code>	Optional: If an encrypted password file is set up, use as the first parameter in the command line to read the password from the full file path and name specified in <i>passwordFile</i> .
<code>/S:server</code>	This parameter is obsolete and values specified with it are ignored. If used, the system checks that the server name specified is not null and that it is greater than 0 characters in length. This parameter is still available to provide backward compatibility. The (default) server for the Outline Load Utility is always localhost.
<code>/A:application</code>	Name of the Oracle Hyperion Planning application to which you are importing.
<code>/U:userName</code>	User name with which to log on to the application.
<code>/CP:commandPropertieFileName</code>	Specifies a file that contains command line arguments that, in conjunction with command line arguments, constitutes the options for execution. For arguments that appear in both the command properties file and the command line, the command line arguments take precedence. See Command Properties File .
<code>/M</code>	Generate fully qualified header records for loadable dimensions in the application. Use <code>/-M</code> if you do not want to display this information (default).
<code>/I:inputFileName</code>	Specifies the data load input file which contains a header record and data records in CSV format. You must also specify a data load dimension (<code>/D</code> option or <code>/TR</code> option). The <code>/ICB</code> switch can be specified to clear Oracle Essbase data.
<code>/IR[:RDBConnectionPropertiesFileName]</code>	Specifies that the input records will come from a relational database source. Specifying the optional properties file designates that some or all of the required relational connection switch properties (<code>/RIQ</code> , <code>/RIC</code> , <code>/RID</code> , <code>/RIR</code> , <code>/RIU</code> , <code>/RIP</code>) can be found within the properties file. A data load dimension (<code>/D</code> option) must be specified as well. The <code>/ICB</code> switch can be specified to clear Essbase data.
<code>/IRA</code>	Identical to the <code>/IR</code> switch except that the required RDB JDBC connection properties (<code>/RIQ</code> , <code>/RIC</code> , <code>/RID</code> , <code>/RIR</code> , <code>/RIU</code> , <code>/RIP</code> switch values) are obtained from the currently connected application's RDB data source. A data load dimension (<code>/D</code> option) must be specified as well. The <code>/ICB</code> switch can be specified to clear Essbase data.
<code>/ICB:blockSpecification</code>	Clears an Essbase block before performing an import operation (<code>/I</code> , <code>/IR</code> , <code>/IRA</code>). (A string of the form <code>"<loadDimensionMembers,...>, <driverMembers,...>, <point-of-view members,...>, <dataLoadCubeName>"</code>).
<code>/ALS</code>	Create alias tables on import if they do not exist (default). Use <code>/-ALS</code> to error out if the referenced alias tables do not exist.


Parameter	Description
<i>/E: outputFileName</i>	Exports the dimension specified with the <i>/D</i> switch to the specified output file. (When exporting planning unit hierarchies, the file is in the format defined for importing planning unit hierarchies.)
<i>/ED: outputFileNameStem</i>	Exports data to Planning Driver Member formatted data files. The <i>/EDD</i> switch must also be set. Files are generated in the form: <code><fileNameStem>.1-n.csv</code> through <code><fileNameStem>.n-n.csv</code> where <i>n</i> is the number of files generated.
<i>/EDH</i>	Exports a dimension header in Planning internal HEADERBLOCK format in the output file. This is used on import to dynamically create base and attribute dimensions prior to import.
<i>/ER[:RDBConnectionPropertiesFileName]</i>	Specifies that export records will be written to a relational database table. Setting the <i>/EDD</i> switch specifies that data will be exported as well. Specifying the optional properties file designates that some or all of the required relational connection switch properties (<i>/REQ</i> , <i>/REC</i> , <i>/RED</i> , <i>/RER</i> , <i>/REU</i> , and <i>/REP</i>) can be found within the properties file. A data load dimension must also be specified using the <i>/D</i> switch.
<i>/ERA</i>	Identical to the <i>/ER</i> switch except that the required RDB JDBC connection properties (<i>/REQ</i> , <i>/REC</i> , <i>/RED</i> , <i>/RER</i> , <i>/REU</i> , <i>/REP</i> switch values) are obtained from the currently connected application's RDB data source. Setting the <i>/EDD</i> switch specifies that data will be exported. A data load dimension must also be specified using the <i>/D</i> switch.
<i>/EDD: dataExportSpecification</i>	Specifies the format of the exported Planning Driver Member formatted data files. A string of the form <code><loadDimensionMembers, ...></code> , <code><driverMembers, ...></code> , <code><point-of-view members, ...></code> , <code><dataLoadCubeName></code> .
<i>/SDM: driverMemberSpecification</i>	Sets driver members on the base load dimension for the application (<i>/A</i>) on import operations only (<i>/I</i> , <i>/IR</i> , <i>/IRA</i>). (A string of the form <code>\ "<baseLoadDimension></code> , <code>\ "<driverMembers, ...>\ "</code> , <code><PlanTypeName>"</code>)
<i>/D: loadDimensionName</i>	Dimension to be loaded, whose member fields correspond to the header record in the load file. You must also specify a load file (<i>/I</i>) or the planning unit hierarchy to be exported with the <i>/E</i> switch. See the following rows to load user-defined dimensions and attributes using <i>/DU</i> , <i>/DA[T]</i> , <i>/DAN</i> , <i>/DAB</i> , and <i>/DAD</i> .
<i>/DU: userDefinedLoadDimensionName</i>	User-defined dimension to be loaded; a dimension with this name will be created if it does not exist.

Parameter	Description
/	Text attribute dimension to be loaded; an attribute dimension with this name, bound to the base dimension, will be created if it does not exist.
DA[T] : <i>attributeLoadDimensionName:baseDimensionName</i>	Text attribute dimension to be loaded; an attribute dimension with this name, bound to the base dimension, will be created if it does not exist.
/	Numeric attribute dimension to be loaded; an attribute dimension with this name, bound to the base dimension, will be created if it does not exist.
DAN : <i>attributeLoadDimensionName:baseDimensionName</i>	Numeric attribute dimension to be loaded; an attribute dimension with this name, bound to the base dimension, will be created if it does not exist.
/	Boolean attribute dimension to be loaded; an attribute dimension with this name, bound to the base dimension, will be created if it does not exist.
DAB : <i>attributeLoadDimensionName:baseDimensionName</i>	Boolean attribute dimension to be loaded; an attribute dimension with this name, bound to the base dimension, will be created if it does not exist.
/	Date attribute dimension to be loaded; an attribute dimension with this name, bound to the base dimension, will be created if it does not exist.
DAD : <i>attributeLoadDimensionName:baseDimensionName</i>	Date attribute dimension to be loaded; an attribute dimension with this name, bound to the base dimension, will be created if it does not exist.
/DX:HSP_Rates	Load the HSP_Rates dimension and create exchange rate tables if they do not exist.
/DS:HSP_SMARTLISTS	Load the Smart Lists dimension and Smart List dimension entries.
/DL:comma tab	Set the field delimiter to the comma character ",", comma (default), or the tab character.
/TR	Load data when driver members are specified in the load file in the Driver Members column. All members except the driver member must be specified in the Point-of-View column. With /TR, you can load one value per row in the load file.
/T	Inherit unspecified plan type settings from the parent when adding new members (default). Use /-T to force explicit setting of plan type settings for the member.
/N	Perform a "dry run" by parsing the load file without loading data or metadata. Use /-N (or do not specify the /N parameter) to parse the load file while loading data and metadata (default).
/O	Maintain the order of members in the load file when loading, with the exception of UDAs (default). Use /-O to ignore the order of members in the load file when loading.



Note:

Performing a dry run parses the load file (for example, checks the header record, checks if the number of values matches the header record number) but does not check the validity of the values defined in the file.

Parameter	Description
/H	Order input records in parent-child order, with the exception of UDAs (default). Use /-H to load input records as they appear in the load file; this option is faster and uses less memory.
/R	Delete all members of the load dimension before performing the load. Use /-R (or do not specify the /R parameter) to keep all members of the load dimension (default). See also /U.
	<div style="border: 1px solid #0070C0; padding: 10px; background-color: #E6F2FF;"> <p> Note: Use caution with /R; this option removes attribute bindings and approvals states.</p> </div>
/DPU	Delete all planning units with the /R parameter, otherwise error-out if members in planning units attempt to be deleted. Use /-DPU to prevent the delete operation from deleting members in planning units.
/IDU	Delete unspecified members not explicitly specified in the load. Members not explicitly specified in the input source will be deleted from the Planning outline on load completion unless: 1) they are an ancestor of a member that was specified, or 2) they are a base member of a shared member that was specified. (/ -IDU is the default.)
/C	Perform a cube refresh after the metadata load. Use /-C if you do not want to perform a cube refresh (default). See also /F.
/F	Create security filters when refreshing with the /C option. Use /-F if you do not want to refresh security filters (default). (This option does not provision users to the application; it only creates security filters for users that currently exist. Users can be provisioned to applications using other methods.) For this option to take effect, /C must also be specified.
/K	Lock the load dimension before loading (default), recommended. Use /-K if you do not want to lock the dimension (not recommended unless you are using /N).
/8	Specifies UTF-8 encoding on input, output, log and exception files and prepends a UTF-8 BOM marker to the output file (default). Use /-8 to not set UTF-8 encoding.

Parameter	Description
/DF:datePattern	<p>Override the default date pattern on date data conversions to the specified pattern. The pattern must be one of the following:</p> <ul style="list-style-type: none"> • MM-DD-YYYY • DD-MM-YYYY • YYYY-MM-DD <p>Use <code>/-DF</code> to use the default date pattern setting (default).</p>
/RIQ:inputQueryOrKey	<p>An SQL query or a key in the command arguments properties file (<code>/CP</code> switch) in which the value is a SQL query that is executed to produce input for an import operation. The <code>/IR</code> switch must be set for this to be used, and if it is used, the <code>/RIC</code>, <code>/RID</code>, <code>/RIR</code>, <code>/RIU</code>, and <code>/RIP</code> switches must also be set.</p>
/RIC:catalog	<p>An RDB JDBC catalog name for the input RDB connection. The <code>/IR</code> switch must be set for this to be used, and if it is used, the <code>/RIQ</code>, <code>/RID</code>, <code>/RIR</code>, <code>/RIU</code>, and <code>/RIP</code> switches must also be set. If the <code>/IRA</code> switch is used, this switch does not need to be specified.</p>
/RID:driver	<p>An RDB JDBC driver name for the input RDB connection. The <code>/IR</code> switch must be set for this to be used, and if it is used, the <code>/RIQ</code>, <code>/RIC</code>, <code>/RIR</code>, <code>/RIU</code>, and <code>/RIP</code> switches must also be set. If the <code>/IRA</code> switch is used, this switch does not need to be specified.</p>
/RIR:url	<p>An RDB JDBC URL for the input RDB connection. The <code>/IR</code> switch must be set for this to be used, and if it is used, the <code>/RIQ</code>, <code>/RIC</code>, <code>/RID</code>, <code>/RIU</code>, and <code>/RIP</code> switches must also be set. If the <code>/IRA</code> switch is used, this switch does not need to be specified.</p>
/RIU:userName	<p>An RDB JDBC username for the input RDB connection. The <code>/IR</code> switch must be set for this to be used, and if it is used, the <code>/RIQ</code>, <code>/RIC</code>, <code>/RID</code>, <code>/RIR</code>, and <code>/RIP</code> switches must also be set. If the <code>/IRA</code> switch is used, this switch does not need to be specified.</p>
/RIP:password	<p>An RDB JDBC password for the input RDB connection. The <code>/IR</code> switch must be set for this to be used, and if it is used, the <code>/RIQ</code>, <code>/RIC</code>, <code>/RID</code>, <code>/RIR</code>, and <code>/RIU</code> switches must also be set. If the <code>/IRA</code> switch is used, the <code>/RIP</code> switch need not be specified.</p> <p>Enter the password in its unencrypted form when specifying it for the first time in the <code>.properties</code> file. When the Outline Load utility is run, the <code>properties</code> file will be rewritten with an encrypted value for the <code>/RIP</code> password. If this value is not specified in the <code>properties</code> file, a command line prompt will be issued to obtain the password.</p>

Parameter	Description
<code>/REQ:exportQueryOrKey</code>	A SQL query or a key in the command arguments properties file (<code>/CP</code> switch) in which the value is a SQL query that specifies the exported values of the form. <code>'INSERT INTO<tableName> (column1, column2, ...) VALUES (property1, property2, ...).'</code> where properties are Planning member properties as found in the flat file column header records. The <code>/ER</code> or <code>/ERA</code> switch must be set for this to be used, and if it is used, the <code>/REC</code> , <code>/RED</code> , <code>/RER</code> , <code>/REU</code> , and <code>/REP</code> switches must also be set.
<code>/REC:catalog</code>	An RDB JDBC catalog name for the export RDB connection. The <code>/ER</code> switch must be set for this to be used, and if it is used, the <code>/REQ</code> , <code>/RED</code> , <code>/RER</code> , <code>/REU</code> , and <code>/REP</code> switches must also be set.
<code>/RED:driver</code>	An RDB JDBC driver name for the export RDB connection. The <code>/ER</code> switch must be set for this to be used, and if it is used, the <code>/REQ</code> , <code>/REC</code> , <code>/RER</code> , <code>/REU</code> , and <code>/REP</code> switches must also be set.
<code>/RER:url</code>	An RDB JDBC URL for the export RDB connection. The <code>/ER</code> switch must be set for this to be used, and if it is used, the <code>/REQ</code> , <code>/REC</code> , <code>/RED</code> , <code>/REU</code> , and <code>/REP</code> switches must also be set.
<code>/REU:userName</code>	An RDB JDBC username for the export RDB connection. The <code>/ER</code> switch must be set for this to be used, and if it is used, the <code>/REQ</code> , <code>/REC</code> , <code>/RED</code> , <code>/RER</code> , and <code>/REP</code> switches must also be set.
<code>REP:password</code>	An RDB JDBC password for the export RDB connection. The <code>/ER</code> switch must be set for this to be used, and if it is used, the <code>/REQ</code> , <code>/REC</code> , <code>/RED</code> , <code>/RER</code> , and <code>/REU</code> switches must also be set. If this value is not specified in the command properties file, a command line prompt will be issued to obtain the password.
<code>/C2A:(column1, alias1), (column2, alias2), ...</code>	"Column to alias" mapping, or runtime column renaming and exclusion. This parameter enables you to rename a column header at runtime, have the column ignored, or have specific plan type properties ignored or renamed.



Note:

An alias specified on the column overrides assignments made by this command.

Parameter	Description
/UCH	Ignore unrecognized column headers and proceed with load. /-UCH is the default current functionality: abort the load if unrecognized column headers are encountered. An information message listing the ignored columns is output.
/X: <i>exceptionFileName</i>	Specify the file that will contain exceptions that occur during the load. (If no file name is specified, the information is written to a file called <code>stderr</code> .)
/L: <i>logFileName</i>	Specify the file that will contain status and informational messages. (If no file name is specified, the information is written to a file called <code>stdout</code> .)
/?	Display usage text.

Example: Load numeric attribute dimension and values, and associate them with the Entity dimension. (An attribute dimension will be created if it does not exist, but no assignment is made of attribute values to base numbers.)

```
OutlineLoad /A:Test /U:admin /M /I:c:/outline1_attribvals_text.csv /
DAN:NumericAttrib:Entity /L:c:/outlineLoad.log /X:c:/outlineLoad.exc
```

```
NumericAttrib,Parent
```

```
One,NumericAttrib
```

```
1,One
```

```
2,NumericAttrib
```

Example: Load Exchange Rates, add EUR as a member of the Currency dimension, and change the year in the load file to match an existing year in the Planning application. The Exchange Rate table is created in the Planning application if it does not exist.

```
OutlineLoad /A:Test /U:admin /M /I:c:/outline1_rates.csv /DX:HSP_Rates /L:c:/
OutlineLogs/outlineLoad.log /X:c:/OutlineLogs/outlineLoad.exc
Table, To Currency, From Currency, Method, Historical, Beg Balance, Year,
Period, Average, Ending
```

```
FX1 , USD, EUR, multiply, 1, 2, FY08, Jan, 3, 4
```

```
FX1 , USD, EUR, , , , FY09, Feb, 5, 6
```

Example: Set Weekly Distribution to Use 445

Account, Parent, Use 445

all,a1,1

Example: Load a file that contains all of the properties available for a UDA. The UDA is loaded and associated with a dimension, but it is not assigned to any member in the dimension.

```
OutlineLoad /A:Test /U:admin /M /I:c:/outline1_uda.csv /D:UDA /L:c:/
OutlineLogs/outlineLoad.log /X:c:/OutlineLogs/outlineLoad.exc
```

Dimension,UDA

Account,New2

Example: Load a file for Currency that does not specify the currency symbol. In this case, the symbol for this currency in the Planning application is set to the ISO symbol, EUR. The scale defaults to 1.

Currency,Parent,Symbol,Scale

EUR,,,

Example: Load a file for Currency that sets the symbol to the name of the new currency. The symbol is automatically set to NewCurr1 in the Planning application for currency NewCurr1. Currency names are limited to 8 characters.

Currency,Parent,Symbol,Scale

NewCurr1,,,

Example: Use the `-f` parameter with an encrypted password

If you have generated an encrypted password file, you can use `-f` as the first parameter in the command line to run the Outline Load utility without entering a password. For example, if you used the `PasswordEncryption` utility to create a password file called `encrypt.txt`, you could use this command line:

```
OutlineLoad -f:c:\encrypt.txt /A:acct /U:admin /M /I:c:/
outline1_accounts.csv /D:Account /L:c:/OutlineLogs/outlineLoad.log /X:c:/
OutlineLogs/outlineLoad.exc
```

Example: `/O` parameter and load file order

In the following load file, if Entity members `e1` and `e2` already exist in the Entity dimension, `e3` could be added as the last sibling, even though it is first in the load file. If `/O` is used, `e3` is

loaded as the first sibling. Because /O is the default, you must specify /-O to have e3 loaded as the last sibling.

```
Entity,Parent,Data Storage,TextAttrib
```

```
e3,Entity,Store,
```

```
e2,Entity,Store,
```

```
e1,Entity,Store,
```

Example: /H parameter and parent/child order

Assume that member e1 already exists, and A and B are new members being loaded. Without /H, an error would display because member B does not exist. With /H, members are sorted internally, so B is loaded first as child of e1, and then A is loaded successfully as child of B.

```
Entity,Parent,Data Storage
```

```
A,B,Store
```

```
B,e1,Store
```

Example: /R parameter

If some members already exist in the dimension, only the members in the input load file should exist in the dimension after the load. If an error occurs during the load after the delete operation, all members of the dimension may be deleted, and the dimension may be empty. Attribute dimensions are not deleted. If a planning unit is started, no Entity members are deleted because the Entity member in the planning unit cannot be deleted.

```
Entity,Parent,Data Storage,TextAttrib
```

```
e1,Entity,Store,
```

```
e11,e1,Store,orange
```

```
e2,Entity,Store,
```

```
e21,e2,Store,
```

```
e11,e2,shared,yellow
```

Example: /T parameter

Load the Account dimension with /T to inherit plan types not explicitly specified in the load file from the parent when adding new members. Assume that member a1 already exists in the application and is valid for all three plan types. After the load completes, member a11 is valid for all three plan types, even though only Plan1 and Plan3 are specified in the load file.

```
Account, Parent, Source Plan Type, Plan Type (Plan1), Plan Type (Plan2), Plan
Type (Plan3)
```

```
a11,a1,Plan1,1,,1
```

Example: /-T parameter

Load the Account dimension with /-T to force explicit setting of plan types for new members. Assume that member a1 already exists in the application and is valid for all three plan types. After the load, member a11 will be valid only for the Plan1 and Plan3 plan types specified in the load file, and not for Plan2.

Example: /TR parameter

```
OutlineLoad /A:acpt1 /U:admin /M /I:c:\outlinedata.csv /TR /L:c:/OutlineLogs/
outlineLoad.log /X:c:/OutlineLogs/outlineLoad.exc
```

```
Value,Driver Member,Point-of-View,Data Load Cube Name
```

```
14,a1,"Jan,Local,e1,Current,Version1,FY08",Plan1
```

```
s11_value2,a2,"Jan,Local,e1,Current,Version1,FY08",Plan1
```

Example: Load Smart List dimensions and Smart List dimension entries using the / DS:HSP_SMARTLISTS parameter.

```
OutlineLoad /A:acpt /U:admin /M /I:c:/smartlist_create1.csv /
DS:HSP_SMARTLISTS /L:c:/OutlineLogs/outlineLoad.log /X:c:/OutlineLogs/
outlineLoad.exc
```

```
SmartList Name, Operation, Label, Display Order, Missing Label, Use Form
Missing Label, Entry ID, Entry Name, Entry Label
```

```
SL1,addsmartlist,SL1Label,,,,,
```

```
SL1,addEntry,,,,,entry1,entrylabel1
```

```
SL1,addEntry,,,,,entry2,entrylabel2
```

Example: Perform incremental data loads using the LINEITEM flag.

You can include a `LINEITEM` flag in the data load file to perform incremental data loads for a child of the data load dimension based on unique driver dimension identifiers. This specifies that data should be overwritten if a row with the specified unique identifiers already exists on the form. If the row does not exist, data is entered as long as enough child members exist under the Data Load Dimension Parent member.

For example, when loading employee data, you can load budget line item detail for predefined Salary Grades. This example shows a command that could be used with a data load file that includes the `LINEITEM` flag.

```
OutlineLoad /A:plnldv /U:admin /M /I:c:\dataload_file.csv /D:"Budget Item"
```

This sample data load file loads data for the Budget Item dimension for children of Grade Changes.

```
"Budget Item","Data Load Cube Name","Point-of-View","Grade Step","Option Value","Start Date","End Date"
```

```
"<LINEITEM("Grade Changes">","HCP","POVMembers","Step1","31721","7/1/09",""
```

```
"<LINEITEM("Grade Changes">","HCP","POVMembers","Step2","32673","7/1/09",""
```

```
"<LINEITEM("Grade Changes">","HCP","POVMembers","Step3","33654","7/1/09",""
```

```
"<LINEITEM("Grade Changes">","HCP","POVMembers","Step4","33654","7/1/09",""
```

In this case, `<LINEITEM("Grade Changes")>` finds the first available member from Budget Item that is a child of the Grade Changes member, based on these unique identifiers selected in the Data Load Settings page: Grade Step, Option Value, Start Date, and End Date.

During data load, if any child members of `Grade Changes` already have data for `Step1` and `7/1/09`, the corresponding member is used to update the remaining data values. If not, the next available empty data row is assigned to `Step1` and `7/1/09`.

When the first data row is processed, the member `Grade1` is assigned. Similarly, the next two members, `Grade2` and `Grade3` are assigned to the second and third data rows. When the fourth data row is processed, `Step1` and `7/1/09` are already assigned to the member `Grade1`, so that row is used to update the value of the remaining fields.

Example: Import a planning unit hierarchy using the `/D` parameter.

```
OutlineLoad /A:acpt /U:admin /I:c:\puh1.csv /D:PUH1
```

When using `/D` to import a planning unit hierarchy, you must specify the name of a planning unit hierarchy (not a dimension). The planning unit hierarchy must already exist in a Planning application before new members can be loaded into it.

Example: Export a planning unit hierarchy using the /E parameter.

```
OutlineLoad /A:acct_580 /U:admin /M /E:puh_test2.csv /D:test2
```

Primary Member, Primary Enabled, Secondary Dimension, Secondary Parent,
Relative Generation, Auto Include, Secondary Member, Include, Owner,
Reviewers, Notifiees

```
e1, true, <none>, <none>, <none>, false, , true, <none>, admin, planner
```

```
e11, true, <none>, <none>, <none>, false, , true, <none>, <none>, <none>
```

```
e2, true, <none>, <none>, <none>, false, , true, <none>, <none>, <none>
```

```
e21, true, Account, a1, 1, false, , true, <none>, <none>, <none>
```

```
e21, true, Account, a1, 1, false, a11, true, <none>, <none>, <none>
```

```
e21, true, Account, a1, 1, false, a12, true, admin, <none>, "admin,admin"
```

```
e21, true, Account, a1, 1, false, a13, true, planner, "planner2,admin", admin
```

```
e21, true, Account, a1, 1, false, a14, true, <none>, <none>, <none>
```

```
e21, true, Account, a1, 1, false, a15, true, <none>, <none>, <none>
```

```
e21, true, Account, a1, 1, false, a16, true, <none>, <none>, <none>
```

```
e21, true, Account, a1, 1-2, false, a111, true, <none>, <none>, <none>
```



Note:

Secondary members for the first four records are not specified.

Dimension Properties

See the following sections for common member properties, and properties specific to Account, Entity, Periods, user-defined dimensions, Year, Scenario, Version, Currency, attribute dimensions, UDAs, exchange rates, Smart Lists, and planning unit hierarchies.

Properties for dynamic members are exposed for all dimensions except Period and Year. The three properties exposed for all dimensions except Period and Year are:

- Enable for Dynamic Children
- Number of Possible Dynamic Children
- Access Granted to Member Creator

Common Member Properties

Member properties common to several dimensions are described in this section. For properties specific to certain dimensions, see the following sections.

- **Parent:** The parent of the member being loaded to create the dimension hierarchy. When you load a member and specify a parent member that is different than the parent member in the application, the member is updated with the new parent value you specified. For example, if Member 1 has a parent value of Member A in your Oracle Hyperion Planning application and you load Member 1 with a parent value of Member B, the system updates your application, making Member B the parent of Member 1. Member 1 and its descendants are moved from Member A to Member B. If you do not specify Parent, it is ignored during the load. The record is rejected if the specified parent is a descendant of the member being loaded, or does not exist in the application.
- **Alias: Default:** Alias defined for the member in the Default Alias table. If you do not specify a value, the alias is not changed in the application. If you specify `<none>` as the value, the alias in the application is deleted.
- **Valid For Consolidations:** Not used by Planning.
- **Hierarchy Type:** Dimensions that are bound to an aggregate storage plan type are automatically enabled to support multiple hierarchies. Specify whether the hierarchy is **Stored** (default), **Dynamic**, or **None**.
- **Data Storage:** The storage attribute for the member being loaded. This value is passed as a string. Default: Never Share. Valid values:
 - **Store**
 - **Dynamic Calc**
 - **Dynamic Calc and Store**
 - **Shared**
 - **Never Share**
 - **Label Only**

Note:

Each plan type can have a different data storage property and data storage properties for each plan type are exposed, for example, Data Storage (Plan1). The original data storage property corresponds to the default data storage. If a data storage property is set to Shared or Label Only, the import will result in an error if any plan type specific storage is set to a value other than Shared or Label Only.

- **Two Pass Calculation:** A Boolean value to indicate whether this attribute is associated with the member being loaded. Use 0 for False and any other number for True. Default: False. For Account members, the Two Pass Calculation property can be set regardless of

the Data Storage setting. For members of dimensions other than Account, the Two Pass Calculation property is valid only when the Data Storage value is Dynamic Calc or Dynamic Calc and Store. Otherwise, the record is rejected.

- **Description:** Description for the member being loaded. If you do not enter a value, new members are loaded without descriptions, and descriptions of existing members are unchanged. If you use <none> as the value, any existing description for that member is deleted.
- **Formula:** Specifies a member formula for the member. By default, there is no member formula associated with a dimension or member. You cannot load member formulas for members that are Shared or Label Only.

 **Note:**

Each plan type can have a different formula and formula properties for each plan type are exposed, for example, Formula (Plan1). The original formula property corresponds to the default formula.

- **UDA:** Specifies the value of the user-defined attributes to bind to the member. Undefined UDAs are added to the dimension. You can add UDAs only to those dimensions that are already created in Planning.
- **Smart List:** Takes the name of a user-defined Smart List that is defined in the application. This value is passed as a string. The default for Smart List is <none>. Only one Smart List can be associated with a member.
- **Data Type:** The data storage value. Valid values:
 - **Currency:** Stores and displays the member value in the default currency.
 - **Non-currency:** Stores and displays the member value as a numeric value.
 - **Percentage:** Stores values as numeric values and displays the member value as a percentage.
 - **Smart List:** Stores values as numeric values and displays the member value as a string.
 - **Date:** Stores and displays the member value in the format mm/dd/yyyy or dd/mm/yyyy.
 - **Text:** Stores and displays the member's value as text.
 - **Unspecified:** Stores and displays the member value as Unspecified.
- **Operation:** Takes these values:
 - **Update:** Adds, updates, or moves the member being loaded.
 - **Delete Level 0:** Deletes the member being loaded if it has no children.
 - **Delete Idescendants:** Deletes the member being loaded and all of its descendants.
 - **Delete Descendants:** Deletes the descendants of the member being loaded, but does not delete the member itself.

Use caution when deleting members; this deletes the member, its data, and any associated planning units.
- **Approvals Enabled:** Enable for approvals. By default, True.
- **Plan Type** (for example, Plan1, Plan2, Plan3): A Boolean value that indicates whether the member being loaded is used in the specified plan. Valid values: 0 for False, or any other

number for True. Default: True. The name depends on the name of the plan type in the application.

- **Aggregation** (Plan1, Plan2, Plan3): The aggregation option for the member being loaded, as related to the specified plan. This is available only if the application is valid for this plan type. This value is passed as a string. Valid values:
 - + (Addition)
 - - (Subtraction)
 - * (Multiplication)
 - / (Division)
 - % (Percent)
 - ~ (Ignore during consolidation)
 - **Never** (Do not aggregate, regardless of hierarchy)
- **UDA**: The value of the UDA being loaded. You can associate UDAs only with dimensions that exist in the application. If a UDA exists, its properties are modified; otherwise, the record is added.

Account Dimension Properties

Account load files can include these properties:

Account, Parent, Alias: Default, Alias: T1, Valid For Consolidations, Data Storage, Two Pass Calculation, Description, Formula, UDA, Smart List, Data Type, Operation, Account Type, Time Balance, Use 445, Use 544, Use 554, Skip Value, Exchange Rate Type, Variance Reporting, Source Plan Type, Plan Type (Plan1), Aggregation (Plan1), Plan Type (Plan2), Aggregation (Plan2), Plan Type (Plan3), Aggregation (Plan3), AttribDim1, AttribDim2

For details on these properties, see the following table and [Common Member Properties](#).

Notes:

- Time Balance specifies how account data flows over time. It takes a type only for members with an account type of Saved Assumption, or if the record is rejected.
- When Time Balance is Flow, records with any valid skip values are loaded, but Skip Value is disabled for all Account types.
- Skip Value can be set when Time Balance is First, Balance, or Average. These options set which values to skip when the parent value is calculated: none, #MISSING, zeros, or #MISSING and zeros.
- Plan type names and numbers depend on what is defined in the application. Base time periods cannot be added with the utility. Year, base time periods, and exchange rates cannot be deleted.
- YearTotal and BegBalance time periods cannot be modified with the utility.
- Exchange Rate Type depends on the value specified for Data Type. Valid values: Average, Ending, and Historical when Data Type is Currency, or None when Data Type is anything other than Currency.
- Variance Reporting loads account members with an account type of Saved Assumption or if the record is rejected. Expense designates the saved assumption as an expense. The actual amount is subtracted from the budgeted amount to determine the variance. Non-Expense designates the Account as not an Expense. The budgeted amount is subtracted from the actual amount to determine the variance. Values for Account types: Revenue:

Non-Expense, Expense: Expense, Asset: Non-Expense, Liability: Non-Expense, Equity: Non-Expense.

- When you update or save the parent of a member, the system verifies if the Source Plan Type associated with the member being loaded is valid for the new parent. If the source plan type of a member is valid for the parent member but not for the member itself, the member is saved, but its source plan type is set to the first valid plan type. If a Source Plan Type is specified but is not valid for the parent, the record is rejected.

Table 5-4 Account Dimension Properties

Column Header in Load File	Oracle Hyperion Planning Property	Value	Default	Required
Account	Member Name	Text, subject to member naming restrictions	None	Yes
Parent	Member Name	Text, subject to member naming restrictions.	None; the name of an existing member, or, if empty, the member is placed as a child under the dimension root	No (Need to specify for most Period members)
Alias:Alias_Table_Name	Alias	Text, subject to Planning member naming restrictions and aliases already defined in the alias table specified in the column header; <none> removes any alias binding for the member from the specified table.	None	No One column header for each alias table defined on the dimension is displayed. Default is a pre-defined alias table defined for every dimension.
Valid For Consolidations	Not Used	N/A	N/A	No
Data Storage	Data Storage	Text: Store, Dynamic Calc and Store, Dynamic Calc, Never Share, Shared, Label Only	Inherited from the parent. If the parent is the root member, the default is Never Share	No
Two Pass Calculation	Two Pass Calculation	True, False, or an integer: non-zero is true; zero is false	Inherited from the parent	No
Description	Description	Text, subject to Planning maximum character restrictions; <none> removes a description	None	No
Formula	Formula	Text, subject to Oracle Essbase restrictions for valid formula syntax; <none> removes a formula	None	No

Table 5-4 (Cont.) Account Dimension Properties

Column Header in Load File	Oracle Hyperion Planning Property	Value	Default	Required
UDA	UDA	A single UDA text value, or a quoted, comma-separated list of UDA text values; non-existing UDAs are added to the dimension; existing UDA bindings are removed on subsequent saves if not re-specified; nothing specified leaves bindings as is; <none> removes all existing UDA bindings.	None	No
Smart List	Smart Lists	Text, restricted to names of Smart Lists that are already defined for the application. Should be specified only if Data Type is set to Smart List; <none> removes any existing Smart List binding	None	No
Data Type	Data Type	Text: Unspecified, Currency, Non-Currency, Percentage, Smart List, Date, Text	Inherited from the Parent; Currency if the member is added under the root dimension	No
Operation		Text: update; delete level 0; delete idescendants; delete descendants	Update	No
Account Type	Account Type	Text: Expense, Revenue, Asset, Liability, Equity, Saved Assumption	Inherited from the parent; Revenue if the member is added under the root dimension.	No
Time Balance	Time Balance	Text: Flow, First, Balance, Average, avg_actual, avg_365, fill	Inherited from the parent (default Time Balance values for Account types: Expense: FlowRevenue: Flow, , Asset: Balance, Liability: Balance, Equity: Balance)	No

Table 5-4 (Cont.) Account Dimension Properties

Column Header in Load File	Oracle Hyperion Planning Property	Value	Default	Required
Use 445 Use 544 Use 554 (Only one column header appears, depending on system settings. If weekly distribution is not defined for the application, no column header displays.)	Weekly Distribution (If Weekly Distribution is set to Even for the application, no distribution option displays.)	True, False, or an integer: nonzero is true; zero is false	None	No
Skip Value		Text: None, Missing, Zeros, Missing and Zeros; must be None if Account Type is Expense or Revenue	Inherited from the parent	No
Exchange Rate Type	Exchange Rate Type	Text: None, Average, Ending, Historical (None should not be specified if Data Type is set to Currency; otherwise, it should be specified)	Inherited from the parent; Average if member is added under root dimension.	No
Variance Reporting	Variance Reporting	Text: Non-Expense, Expense. (Must be Expense if Account Type is Expense; must be Non-Expense for all other Account types.	Inherited from the parent; Non-Expense if the member is added under the root dimension.	No
Source Plan Type	Source Plan Type	Text: Plan Type names defined in the application (for example, Plan1 or Plan2)	Plan1 or the name of the first plan type defined in the application	No
Plan Type (<i>Plan1</i>)	Plan Type	True, False, or an integer: non-zero is true; zero is false	Inherited from the parent	No
Aggregation (<i>Plan1</i>)	Aggregation	Text: +, -, *, /, %, ~, Never	Inherited from the parent. If the parent is a root member, the default is +; for Year, the default is ~ (ignore)	No
Plan Type (<i>Plan2</i>)	Plan Type	True, False, or an integer: non-zero is true; zero is false	Inherited from the parent	No
Aggregation (<i>Plan2</i>)	Aggregation	Text: +, -, */%~, Never	Inherited from the parent; if the parent is the root member, the default is +; for Year, the default is ~ (ignore)	No
Plan Type (<i>Plan3</i>)	Plan Type	True, False, or an integer: non-zero is true; zero is false	Inherited from the parent	No

Table 5-4 (Cont.) Account Dimension Properties

Column Header in Load File	Oracle Hyperion Planning Property	Value	Default	Required
Aggregation (<i>Plan3</i>)	Aggregation	Text: +, -*/%, ~, Never	Inherited from the parent; if the parent is a root member, the default is +; for Year, the default is ~ (ignore)	No
Attribute Dimension Name (this property is available for sparse Account, Entity, or user-defined dimensions)	Attributes	The name of an attribute defined in the attribute dimension: existing attribute bindings are removed on subsequent saves if not re-specified; nothing specified leaves bindings as is; <none> removes all existing attribute bindings for the member; one column header is displayed for each Attribute dimension defined on the dimension.	None	No

Entity Dimension Properties

Entity, Parent, Alias: Default, Alias: T1, Valid For Consolidations, Data Storage, Two Pass Calculation, Description, Formula, UDA, Smart List, Data Type, Operation, Base Currency, Plan Type (*Plan1*), Aggregation (*Plan1*), Plan Type (*Plan2*), Aggregation (*Plan2*), Plan Type (*Plan3*), Aggregation (*Plan3*), *AttribDim1*, *AttribDim2*

Notes:

- **Entity:** The Entity information being loaded.
- **Base Currency:** Displayed only for multicurrency applications. Takes the code of the currency for the Entity being loaded, as defined in the application.
- Plan type names and numbers depend on what is defined in the application.

Table 5-5 Entity Dimension Properties

Column Header in Load File	Oracle Hyperion Planning Property	Value	Default
Entity	Member Name	Text, subject to Planning member naming restrictions.	None
Two Pass Calculation	Two Pass Calculation	True, False, or an integer: non-zero is true; zero is false (should be set to 1 only if Data Type is Dynamic Calc or Dynamic Calc and Store)	Inherited from the parent

Table 5-5 (Cont.) Entity Dimension Properties

Column Header in Load File	Oracle Hyperion Planning Property	Value	Default
Base Currency	Base Currency	Text, restricted to currency names already defined in the application	Inherited from the parent; if the member is shared, the default is the base member's currency; if the member is added under the root dimension, the base currency is the default currency defined when the application was created.

Period Dimension Properties

Period, Parent, Alias: Default, Alias: T1, Data Storage, Two Pass Calculation, Description, Formula, UDA, Smart List, Data Type, Operation, Type, Start Period, End Period, Aggregation (Plan1), Aggregation (Plan2), Aggregation (Plan3)

Notes:

- For Period, Parent should be specified for most update operations.
- Type (such as Base or Rollup) cannot be changed for existing members. You can add and update Rollup and Alternate members. You can update BegBalance and Base time periods (for example, to add an alias). You cannot add or modify YearTotal time period. The DTS Period type is recognized, but is not supported by the Outline Load utility. If you try to load DTS information, the utility displays an error in the log file.
- Start Period and End Period are valid for Rollup Period types.
- Plan type names and numbers depend on what is defined in the application.
- For properties common to Period and Account members, see [Account Dimension Properties](#).

Table 5-6 Period Dimension Properties

Column Header in Load File	Oracle Hyperion Planning Property	Value	Default	Required
Period	Member Name	Text, subject to Planning restrictions on Period member names.		
Type	Not exposed in Planning	Text: base, rollup, year, alternate, DTS; only Rollup and alternate time periods can be loaded by the utility; BegBalance and Base periods can be modified (for example, to add an alias); YearTotal and DTS time periods cannot be loaded or modified.	None	Yes

Table 5-6 (Cont.) Period Dimension Properties

Column Header in Load File	Oracle Hyperion Planning Property	Value	Default	Required
Start Period	Text (only applicable to summary time periods)	Text, restricted to time periods already defined for the application.	None	Yes
End Period	Text (only applicable to summary time periods)	Text, restricted to time periods already defined for the application.	None	Yes

User-Defined Dimension Properties

User Defined Dimension Name, Parent, Alias: Default, Alias: T1, Valid For Consolidations, Data Storage, Two Pass Calculation, Description, Formula, UDA, Smart List, Data Type, Operation, Aggregation (Plan1), Aggregation (Plan2), Aggregation (Plan3)

Plan type names and numbers depend on what is defined in the application. For properties common to User-Defined and Account members, see [Account Dimension Properties](#).

Table 5-7 User-Defined Dimension Properties

Column Header in Load File	Oracle Hyperion Planning Property	Value	Default	Required
<i>User-defined dimension name</i>	Member Name	Text, subject to Planning member naming restrictions. Name of the user-defined member being added to the dimension.	None	Yes
Parent	Member Name or root Dimension Name	Text, subject to Planning member naming restrictions.	Root dimension member	Yes

Year Dimension Properties

Year, Parent, Alias: Default, Alias: T1, Data Storage, Two Pass Calculation, Description, Formula, UDA, Smart List, Data Type, Operation

Notes:

- The member name for Year must be of the form *FYnn*.
- If the last existing year is less than the year value entered, the intervening years are also created. For example, if the last defined year is FY08 and you enter FY11, the utility creates FY09, FY10, and FY11.

Table 5-8 Year Dimension Properties

Column Header in Load File	Oracle Hyperion Planning Property	Value	Default	Required
Year	Year member	<i>FYnn</i>	None	Yes

Scenario Dimension Properties

Scenario, Parent, Alias: Default, Alias: T1, Valid For Consolidations, Data Storage, Two Pass Calculation, Description, Formula, UDA, Smart List, Data Type, Operation, Start Year, Start Period, End Year, End Period, Exchange Table, Include BegBal, Approvals Enabled, Aggregation (Plan1), Aggregation (Plan2), Aggregation (Plan3)

Notes:

- No Year cannot be selected as Start Year or End Year. BegBalance cannot be selected as Start Period or EndPeriod.
- If not specified, Start Year and End Year and Start Period and End Period are set to default values (first and last year in the application, and first and last base period in the application).
- Plan type names and numbers depend on what is defined in the application.

Table 5-9 Scenario Dimension Properties

Column Header in Load File	Oracle Hyperion Planning Property	Value	Default	Required
Scenario	Scenario Name	Text	None	Yes
Start Year		First FY year defined in the application (as determined by position)	First year in the application	No
Start Period		First base time period (as determined by position)	First base period in the application	No
End Year		Last FY year defined in the application (as determined by position)	Last year in the application	No
End Period		Last base time period (as determined by position)	Last base period in the application	No
Exchange Table		The name of an exchange rate table defined in the application.	None	No
Include BegBal		True, False, or an integer: non-zero is true; zero is false	False	No
Approvals Enabled		True, False, or an integer: non-zero is true; zero is false	False	No

Version Dimension Properties

Version, Parent, Alias: Default, Alias: T1, Data Storage, Two Pass Calculation, Description, Formula, UDA, Smart List, Data Type, Operation, Version Type, Approvals Enabled, Aggregation (Plan1), Aggregation (Plan2), Aggregation (Plan3)

Plan type names and numbers depend on what is defined in the application.

Table 5-10 Version Dimension Properties

Column Header in Load File	Oracle Hyperion Planning Property	Value	Default	Required
Version	Version Name	Text	None	Yes
Version Type	Version Type	Bottom Up or Target	Bottom Up	No
Approvals Enabled	Enabled for Approvals	True, False, or an integer: non-zero is true; zero is false	False	No

Currency Dimension Properties

Currency, Parent, Alias: Default, Alias: T1, Data Storage, Two Pass Calculation, Description, Formula, UDA, Smart List, Data Type, Operation, Symbol, Scale, Triangulation Currency, Reporting Currency, Thousands Separator, Decimal Separator, Negative Style, Negative Color

If the Currency symbol is not specified, it is set by default to the ISO symbol if the currency being loaded is defined in Oracle Hyperion Planning.

Table 5-11 Currency Dimension Properties

Column Header in Load File	Planning Property	Value	Default	Required
Currency	Currency Name	Text	None	Yes
Symbol	Symbol	Text, subject to Planning currency symbol naming restrictions	ISO symbol if the currency being loaded is defined Planning; the same as the currency name if the currency being loaded is not defined in Planning	No
Scale	Scale	An integer value from 0 to 9, where 0 corresponds to 1, 1 corresponds to 10, 2 corresponds to 100, and so on	No scaling	No
Triangulation Currency	Triangulation Currency	A currency defined in the application	None	No
Reporting Currency	Reporting Currency	True, False, or an integer: non-zero is true; zero is false	False	No
Thousands Separator	Thousands Separator	default; none; comma; dot; space	None	No
Decimal Separator	Decimal Separator	default; dot; comma	dot	No
Negative Style	Negative Sign	default; prefixed; suffixed; parentheses	prefixed	No
Negative Color	Negative Color	default; black; red	black	No

Attribute Dimension Properties

Attribute, Parent, Alias: Default, Operation

Notes:

- For properties common to Attribute and Account members, see [Account Dimension Properties](#).
- **Custom attributes:** You can load attribute values to the attribute dimension for text, numeric, Boolean, and date attributes. If you modify properties and do not specify a value, the custom attribute is not changed in the application. To remove a custom attribute, specify <none> as the value. The value is passed as a string.
 - **Update:** Adds, updates, or moves the member that is being loaded.
 - **Delete Level 0:** Deletes the member that is being loaded if it has no children.
 - **Delete Idescendants:** Deletes the member that is being loaded and all of its descendants.
 - **Delete Descendants:** Deletes the descendants of the member that is being loaded, but does not delete the member itself.

Use caution when deleting members; this deletes the member, its data, and any associated planning units.

Table 5-12 Attribute Dimension Properties

Column Header in Load File	Oracle Hyperion Planning Property	Value	Default	Required
Attribute	Attribute Value Name	Text, subject to Planning member naming restrictions	None	Yes

UDA Dimension Properties

Dimension, UDA, Operation

For properties common to UDA and Account members, see [Account Dimension Properties](#).

Table 5-13 UDA Dimension Properties

Column Header in Load File	Oracle Hyperion Planning Property	Value	Default	Required
Dimension	Name of base dimension for which UDA is defined	Text, name of dimension for which the UDA will be defined; UDAs cannot be defined for Attribute dimensions	None	Yes
UDA	UDA being defined	Text, subject to Planning member naming restrictions	None	Yes

Exchange Rate Dimension Properties

Table, Description, To Currency, From Currency, Method, Historical, Beg Balance, Year, Period, Average, Ending

Table 5-14 Exchange Rate Dimension Properties

Column Header in Load File	Oracle Hyperion Planning Property	Value	Default	Required
Table	fxTblId	Name of the exchange rates table	None	Yes
Description	Description	Description of the exchange rates table	None	No
To Currency	toCur	Currency defined in the application, to which the conversion will be applied	None	Yes
From Currency	fromCur	Currency defined in the application, from which the conversion will be computed	None	Yes
Operation	N/A	update (delete operations are not supported: delete level 0, delete idescendants, delete descendants)	update	No
Method	method	multiply; divide	multiply	No
Historical	historicalRate	numeric value	0	No
Beg Balance	begBalanceRate	numeric value	0	No
Year	yearId	A year defined in the application, such as FY08	None	Yes, if Average or Ending is specified
Period	tpId	A base time period defined in the application, such as Jan	None	Yes, if Average or Ending is specified
Average	avgVal	numeric value	None	No
Ending	endVal	numeric value	None	No

Planning Unit Hierarchy Dimension Properties

Primary Member, Primary Enabled, Secondary Dimension, Secondary Parent, Relative Generation, Auto Include, Secondary Member, Include, Owner, Reviewers, Notifiees

Table 5-15 Planning Unit Hierarchy Dimension Properties

Column Header in Load File	Oracle Hyperion Planning Property	Value	Default	Required
Primary Member	Primary dimension (Entity) member name. This property corresponds to the Name column of the Primary and Subhierarchy Selection tab of the Planning Unit Hierarchy page.	Text	None	Yes
Primary Enabled	Enables the Primary Member for Approvals. This property corresponds to the Enabled column of the Primary and Subhierarchy Selection tab of the Planning Unit Hierarchy page.	True, False, or an integer: non-zero is true; zero is false	None	No
Secondary Dimension	Dimension name that specifies the Subhierarchy Dimension to be associated with the Primary Member. This property corresponds to the Dimension column of the Primary and Subhierarchy Selection tab of the Planning Unit Hierarchy page.	Text	None	No
Secondary Parent	Dimension member name that is referenced in specifying Secondary Members with the Relative Generation property. This property corresponds to the Parent Member column of the Primary and Subhierarchy Selection tab of the Planning Unit Hierarchy page.	Text	None	No

Table 5-15 (Cont.) Planning Unit Hierarchy Dimension Properties

Column Header in Load File	Oracle Hyperion Planning Property	Value	Default	Required
Relative Generation	A numeric range specifying the descendants of the Secondary Parent for participation in Approvals depending on the Auto Include and Include properties. This property corresponds to the Relative Generation column of the Primary and Subhierarchy Selection tab of the Planning Unit Hierarchy page.	Integers specifying a range of generations. For example, 1-3 designates generations {1,2,3}; 1, 3-5 designates generations {1,3,4,5}; 0-4, 7 designates generations {0,1,2,3,4,7}, and so on. Level 0 is equal to the generation of the Secondary Parent, level 1 is equal to the generation of immediate children of the Secondary Parent, and so on.	None	No
Auto Include	Enables for Approvals the descendant members specified with the Secondary Parent and Relative Generation properties. This setting can be overridden on a member-by-member basis with the Include property. This property corresponds to the Auto Include check box on the Primary and Subhierarchy Selection tab of the Planning Unit Hierarchy page.	True, False, or an integer: non-zero is true; zero is false	None	No
Include	Enables for Approvals the specified subhierarchy member. This property corresponds to the check boxes in the Selected Members pop-up window that displays when the icon on the far right column of the Primary and Subhierarchy Selection tab of the Planning Unit Hierarchy page is clicked.	True, False, or an integer: non-zero is true; zero is false	True	No

Table 5-15 (Cont.) Planning Unit Hierarchy Dimension Properties

Column Header in Load File	Oracle Hyperion Planning Property	Value	Default	Required
Secondary Member	<p>Secondary dimension member name that was enabled for Approvals. The Include, Owner, Reviewers, and Notifiees properties are dependent on this property.</p> <p>This property corresponds to the Selected Members column of the Primary and Subhierarchy Selection tab, and the Name column (where Secondary Members are specified) of the Assign Owners tab, of the Planning Unit Hierarchy page.</p>	Text	None	Yes, if these column headers are defined: Owner, Reviewers, Notifiees
Owner	<p>User name of the owner for the specified member.</p> <p>This property corresponds to the Owner column of the Assign Owners tab of the Planning Unit Hierarchy page.</p>	Text	None	No

Table 5-15 (Cont.) Planning Unit Hierarchy Dimension Properties

Column Header in Load File	Oracle Hyperion Planning Property	Value	Default	Required
Reviewers	A comma separated list of reviewers for the specified member. This property corresponds to the Reviewer column of the Assign Owners tab of the Planning Unit Hierarchy page.	Text	None	No


 **N**
o
t
e
:
S
p
e
c
i
f
y
t
h
e
r
e
v
i
e
w
e
r
s
i
n
t
h
e
o
r
d
e
r
y
o
u
w

Table 5-15 (Cont.) Planning Unit Hierarchy Dimension Properties

Column Header in Load File	Oracle Hyperion Planning Property	Value	Default	Required
		a n t h e m t o r e v i e w t h e p l a n n i n g u n i t . T h e f i r s t r e v i e w e r i n t h e l		

Table 5-15 (Cont.) Planning Unit Hierarchy Dimension Properties

Column Header in Load File	Oracle Hyperion Planning Property	Value	Default	Required
		i s t i s t h e f i r s t u s e r t o w o r k o n t h e p l a n n i n g u n i t . W h e n t h e f i r s t		

Table 5-15 (Cont.) Planning Unit Hierarchy Dimension Properties

Column Header in Load File	Oracle Hyperion Planning Property	Value	Default	Required
		user prompts to step through planning unit, these conditions review in the list		

Table 5-15 (Cont.) Planning Unit Hierarchy Dimension Properties

Column Header in Load File	Oracle Hyperion Planning Property	Value	Default	Required
		t b e c o m e s t h e p l a n n i n g u n i t o w n e r , a n d s o n t h r o u g h t h e l i s t o f r		

Table 5-15 (Cont.) Planning Unit Hierarchy Dimension Properties

Column Header in Load File	Oracle Hyperion Planning Property	Value	Default	Required
		event source create		
Notifiees	A comma separated list of Notifiees for the specified member. This property corresponds to the Notify Users column of the Assign Owners tab of the Planning Unit Hierarchy page.	Text	None	No

Smart List Dimension Properties

SmartList Name, Operation, Label, Display Order, Missing Label, Use Form Setting, Entry ID, Entry Name, Entry Label

Table 5-16 Smart List Dimension Properties

Column Header in Load File	Oracle Hyperion Planning Property	Value	Default	Required
SmartList Name	Name of the Smart List	Text, name of the Smart List/enumeration	None	Yes
Operation	Operation	Text, addsmartlist—creates a new Smart List; addEntry—adds an entry to the Smart List	None	Yes
Label	Label field of Smart List	Text, label field of Smart List	Empty	No

Table 5-16 (Cont.) Smart List Dimension Properties

Column Header in Load File	Oracle Hyperion Planning Property	Value	Default	Required
Display Order	Display Order	Integer or text value for display order of the Smart List: 0 or ID—order by entry id; 1 or Name—order by entry name; 2 or Label—order by entry label	ID	No
Missing Label	#Missing Drop Down label	Text, Missing Drop Down label	LABEL_NONE	No
Use Form Setting	#Missing Form	True—Form Setting; False—Drop Down Setting	Form Setting	No
Entry ID	Smartlist entry ID	ID for the Smart List entry	Default entry ID	No
Entry Name	Entry Name	N/A	None	Yes
Entry Label	Entry Label	N/A	None	Yes

Working with Planning Import and Export

Use Oracle Hyperion Planning Import and Export to perform import and export operations without using a command line. Administrators can import metadata and data for Account, Period, Year, Scenario, Version, Currency, Entity, user-defined dimensions, attributes, UDAs, exchange rates, Smart Lists, and planning unit hierarchies from a flat file. You can also export metadata and data to a flat file using Planning Import and Export.

Note:

Planning Import and Export does not support a relational import or export of data or metadata. To import or export metadata or data to or from a relational data source, use the Outline Load utility. See [Working with the Outline Load Utility](#).

Running Planning Import and Export

To import or export data or metadata using Oracle Hyperion Planning Import and Export:

1. Back up the application and application databases before loading information. See the *Oracle Enterprise Performance Management System Backup and Recovery Guide*.
2. **Optional:** If importing, create a command properties file (`.properties`). See [Command Properties File](#).
3. If importing, generate a load file for each dimension or set of data that you want to load. See [Generating Load Files](#).
4. Log in to the Planning application.
5. Select **Administration**, then **Import and Export**, and then select a task:

- To import metadata from a flat file, select **Import Metadata from File**.
For option descriptions, see [Importing Metadata from a File](#).
 - To import data from a flat file, select **Import Data from File**.
For option descriptions, see [Importing Data from a File](#).
 - To export metadata to a flat file, select **Export Metadata to File**.
For option descriptions, see [Exporting Metadata to a File](#).
 - To export data to a flat file, select **Export Data to File**.
For option descriptions, see [Exporting Data to a File](#).
6. Click **Run**.
 7. **Optional:** Clear any temporary files that were generated during the import or export. See [Clearing EPM Oracle Instance Tmp Directory Files](#).

When importing data or metadata, you have the option of running the import in the background. Click **Run In Background** on the progress bar to enable background processing.

When exporting data, if the data export exceeds three minutes in duration, the following message will display and the export will automatically run in the background:

```
"The export has exceeded the maximum configured wait time and will now be performed in the background. To check export status, open the Import and Export Status page. Note that exported data files should be downloaded and saved because they may be periodically removed from temporary locations on the server."
```

Once this message displays, clicking **View Status** or selecting **Tools**, and then **Import and Export Status** will display a **Processing** status in the **Status** column. When the data export is complete, the **Status** column will display a **Complete** status. Selecting the **Complete** status will launch the **View Status** window and display a green arrow. Click the green arrow to download the exported file from a temporary location and save it to a permanent location. If the file in the temporary location no longer exists, the following message is displayed:

```
"The file that you tried to download no longer exists on the server. Export your data again."
```



Note:

You cannot run the Export Metadata operation in the background.

Importing Metadata from a File

To import an attribute, user-defined dimension, Smart List, or exchange rates, select **New Dimension** in the **Dimension** drop down, and then select the type in **Dimension Type**.

In **Import Metadata from FlatFile**, select the following options:

Table 5-17 Import Metadata from File Options

Option	Description
Dimension to Import	<p>Dimension to be loaded, whose member fields correspond to the header record in the metadata file. If needed, you can create a new dimension on import.</p> <p>User-defined dimensions (for example, Product) are listed in the drop-down list along with the base dimensions. If there is a user-defined dimension in the application and you want to load its metadata, then select the user-defined dimension from the drop-down list. If your file contains metadata for a user-defined dimension called, for example, NewProduct, which does not yet exist in the application, then select New Dimension, then User Defined, and then NewProduct. Oracle Hyperion Planning Import and Export will create the NewProduct dimension and load its metadata.</p>
Metadata File to Import	<p>Specify the metadata load input file which contains a header record and metadata records.</p> <p>See Loading Metadata.</p>
Create Security Filters	<p>Creates security filters when performing a cube refresh. Deselect this option if you do not want to refresh security filters. (This option does not provision users to the application; it only creates security filters for users that currently exist. Users can be provisioned to applications using other methods.)</p>
Database Refresh	<p>Performs a database refresh after the metadata load.</p>
Delete All Planning Units	<p>Deletes all planning units or displays an error if members in planning units would be deleted. Select this option along with the Before Importing, Delete All Members from the Dimension option to enable deleting started planning units and deleting all members in the dimension specified in the load file.</p>
Before Importing, Delete All Members from the Dimension	<p>Deletes all members of the load dimension before performing the load. See also Delete All Planning Units. Clear this option to keep all members of the load dimension.</p>



Note:

Use caution. This option removes attribute bindings and approvals states.

Table 5-17 (Cont.) Import Metadata from File Options

Option	Description
Assume Parent's Plan Types if Unspecified in File	Inherits unspecified plan type settings from the parent when adding new members. Clear this option to force explicit setting of plan type settings for the member.
Lock Load Dimension before Load	Locks the load dimension before loading. Clear this option if you do not want to lock the dimension (not recommended unless you are performing a dry run).
Use the Member Order in the Import File	Maintains the order of members in the load file when loading, with the exception of UDAs. Clear this option to ignore the order of members in the load file when loading.
Sort Import File in Parent-Child Order	Orders input records in parent-child order, with the exception of UDAs. Clear this option to load input records as they appear in the load file. Clearing this option is faster and uses less memory.
Dry Run	Performs a "dry run" by parsing the load file without loading data or metadata. Clear this option to parse the load file while loading data and metadata.

**Note:**

Performing a dry run parses the load file (for example, checks the header record, checks if the number of values matches the header record number) but does not check the validity of the values defined in the file.

Importing Data from a File

To import data, the load dimension must be selected in the **Dimension** list.

Table 5-18 Import Data from File Options


Option	Description
Data Import Format	<p>Choose a data import file format:</p> <ul style="list-style-type: none"> • Planning—Oracle Hyperion Planning file format
	<div style="border: 1px solid #0070C0; padding: 10px;"> <p> Note:</p> <p>If data is exported to a file in Planning format, you can use Notepad to change the database name in the file; for example, ASOCube, and then import the data file into the aggregate storage outline provided all dimensions exist in the aggregate storage outline.</p> </div>
	<ul style="list-style-type: none"> • Essbase—Oracle Essbase file format. (exported by an on-premises product). These files can be formatted in either Cell list or Column list format. Cell list format specifies a single cell value for each record, whereas Column list format specifies multiple cell values for each record. For example, if you specify Time on the column, you might have the values Jan, Feb, Mar, and so forth on the column, with 12 values for each record (that is, one for each month). Conversely, the Cell list equivalent would contain 12 records, one for each month. For information on exporting files from Essbase, see the on-premises <i>Oracle Essbase Administration Services Online Help</i>.
Dimension to Import	<p>For a Planning data import format, select the dimension into which to load data, whose member fields correspond to the header record in the data file. This dimension must exist in the application.</p>
Plan Type	<p>For an Essbase data import format, select the plan type into which to import the data.</p>
Data File to Import	<p>Specify the data load input file which contains a header record and data records.</p>
Driver Member Specification	<p>For a Planning data import format, enter the driver members specified for the import dimension. If you leave this field empty and you specified import and driver members on the Data Load Administration window, those settings are used for the import. Conversely, values that you enter here override those on the Data Load Administration window.</p>
Date Format	<p>Choose an import date format:</p> <ul style="list-style-type: none"> • MM-DD-YYYY • DD-MM-YYYY • YYYY-MM-DD

Table 5-18 (Cont.) Import Data from File Options

Option	Description
Clear Essbase Data	<p>Clears an Essbase block before performing an import.</p> <p>For example:</p> <pre>e2, IDescendants (Account_Total), "FY13, Current, BU Version_1, Jan, Local", Plan1</pre>

 **Note:**

If a data value exists in a cell and the imported data does not have a value for that cell, the existing data is not overwritten. To overwrite existing data with a #missing value, the data block must be cleared first (all cells will be #missing) and then data should be imported into these cells.

Exporting Metadata to a File

In **Export Metadata to File**, select the following options:

Table 5-19 Export Metadata to File Options

Option	Description
Dimension	The dimension you are exporting

After clicking **Run**, the browser prompts you to either open the file or save it to your local computer.

Exporting Data to a File

Describe the form that contains the data being exported. You must enter the plan type, row, column, and POV member information that was entered during form design.

When entering member names with a comma character in the **Export Data to File** window, use double quotation marks to enclose the member names. Each member in a list of multiple members must be placed in its own set of quotation marks. For example:

```
"e2, ""e,2"""  
"Jan, ""Feb,2"" ,Mar"  
"FY13,Current, ""a,1A"" ,Local, BU Version_1"
```

In **Export Data to File**, select the following options:

Table 5-20 Export Data to File Options

Option	Description
Plan Type	The plan type that was entered during form design
Row Members	The row information that was entered during form design Example: Idescendants (Ent_Total)
Column Members	The column information that was entered during form design Example: Idescendants (YearTotal)
POV Members	The Page dimensions and member information Example: FY13,Current, BU Version_1,account1,Local

Clearing EPM Oracle Instance Tmp Directory Files

Each time an import or export occurs, Oracle Hyperion Planning generates temporary files which are placed in the EPM Oracle Instance tmp directory. These files are not deleted, and over time, they can build up and affect performance. Oracle recommends that you periodically clear the EPM Oracle Instance tmp directory.

For more information about EPM Oracle Instance, see [About EPM Oracle Instance](#).

Enabling Data Load of Essbase Data

You specify parameters to enable data to be loaded directly into an Oracle Essbase database. Optionally, you can use advanced settings if you want to load details to child members of parent dimension members based on unique identifiers for the driver dimension.






For example, a company might load the Employee dimension member with account data for Start Date, Position, Salary Basis, and Pay Type. Because the human resource data includes placeholders for new and existing employees, the company could set up the following advanced settings:

- Data load dimension parents: New Employees, Existing Employees
- New Employees unique identifiers: Start Date, Position
- Existing Employees unique identifiers: Salary Basis, Pay Type

During data load, child members of New Employees and Existing Employees are evaluated for data updates. The unique identifiers Start Date, Position, Salary Basis, and Pay Type determine if existing data load dimension values are updated, or if new values are added: If the

unique identifier's data values are the same, data is updated. If the data value is different, the next available child member is used.

To specify parameters for loading data:

1. Select **Administration**, then **Data Load Settings**.
2. For **Data Load Dimension**, select the dimension (such as Employee) for which data is loaded for the application.
The dimension corresponds to the information to be loaded.
3. For **Driver Dimension**, click  to select the dimension into which data is loaded.
For example, if you are loading data to Employee, the driver dimension might be Account.
4. Select the members of the driver dimension.
For example, if the driver dimension is Account, driver dimension members might include Start Date, Grade, Position, Salary Basis, and Pay Type.
5. **Optional:** To use advanced settings, complete these steps.
 - a. Add a row by clicking  above **Data Load Dimension Parent**,
 - b. To the right of the new field, click  and select a parent member.
For information about selecting members, see [Using the Member Selector](#).
 - c. To the right of the parent member, under **Driver Dimension Unique Identifiers**, select one or more members as unique identifiers. (Members selected for this field must be included in the list of selected Driver Dimension members at the top of the page.)
Each parent member must include at least one unique identifier member. These members determine if existing data load dimension values are updated, or if new values are added.
 - d. If necessary, continue adding rows by repeating step 5.a through step 5.c.
 - e. To duplicate or delete a row, select the check box to the left of the row, and click  or . You can duplicate only one selected row at a time.
6. Click **Save**.

Loading from Administration Services

To load from Oracle Essbase Administration Services:

1. Open Administration Services Console.
2. See the Oracle Essbase documentation for instructions on connecting to the server and entering a username and password.
3. Minimize the Administration Services Console.
4. Open Windows NT Explorer (**Start**, then **Programs**, and then **Windows NT Explorer**).
5. Browse to the directory containing data files to load.
6. Select text files to load and drag them onto Administration Services Console on the Windows Task Bar at the bottom of the screen.
7. Continue holding the mouse while dragging files onto the Administration Services Console window, then release it.

8. See the Essbase documentation for instructions on selecting the application to which to load data files.
9. Select the database to which to load data files.
Select the database corresponding to the plan type in the Oracle Hyperion Planning application into which you are loading data.

File Example

This example loads data values for the first quarter of 2008 for the Europe entity and the Gross Sales account in the Actual scenario and Final version.

EUROPE	GROSS SALES	ACTUAL	FINAL	JAN	2008	150
EUROPE	GROSS SALES	ACTUAL	FINAL	FEB	2008	110
EUROPE	GROSS SALES	ACTUAL	FINAL	MAR	2008	200

If a dimension uses the same value for all rows, you can place it in the header as a page dimension, as in this example:

ACTUAL	FINAL			
EUROPE	GROSS SALES	JAN	2008	150
ASIA	NET INCOME	FEB	2008	150
EUROPE	NET INCOME	FEB	2008	110
ASIA	GROSS SALES	JAN	2008	200

6

Managing Forms

Related Topics

- [About Forms](#)
- [Creating Simple Forms](#)
- [Creating Composite Forms](#)
- [Designing Specific Types of Forms](#)
- [Working with Forms and Form Components](#)

About Forms

Forms are grids for entering data. You can create simple forms or composite forms to meet your needs. Because composite forms consist of simple forms, you must create simple forms before creating composite forms.

Form Components

Point of View

Select members for the point of view to determine the context for pages, rows, and columns. For example, if the Scenario dimension is set to Budget in the point of view, all data entered in pages, rows, and columns goes into the Budget scenario. The point of view is set to one member, which a user cannot change, for each point of view dimension.

To simplify the form, in the point of view you can specify only relevant members or include user variables. See [Defining Simple Form Page and Point of View](#) and [Managing User Variables](#).

Page Axis

Use the page axis to specify combinations of members that may span dimensions so users can work with data in smaller, logical views. Each item on the page axis can have members selected from one or more dimensions. Users see only members they can access.

You can specify multiple page drop-down lists, and select members using relationship functions or attributes. Switch between member sets by selecting them from the page axis.

You display member names or aliases on the page axis. You can specify the number of members in a page dimension that enables a search drop-down list on the data entry page, useful if dimensions contain many members. See [Defining Simple Form Page and Point of View](#).

Rows and Columns

Rows and columns define the grid into which users enter data. For example, you can assign Unit Sales to the row axis and January to the column axis. When users access forms, they can enter data into the cell where the Unit Sales row intersects with the January column.

By default, forms have one set of rows and columns. You can add rows and columns to create asymmetrical combinations of members. See [Creating Asymmetric Rows and Columns](#).

Form Design Considerations

Forms and Plan Types

When you create a form, you associate it with a plan type, which determines the form's valid members. For example, if you assign a form to the Revenue plan type, you can add only accounts that are valid for the Revenue plan type. Entered data is saved to the selected plan type's database. You cannot change the plan type for a form after assigning it.

You can edit form accounts if their source plan type matches the form's plan type. If you add an account to a form for a plan type other than the account's source plan type, the account is read-only on that form.

Forms and Access Permissions

By assigning access to a form, you control which users can change its design (for example, its layout and instructions) and input data. Users can select only members to which they have read or write access. Users can edit forms only if they have access to at least one member of each secured dimension. For example, if users have read-only access to the Europe entity, the rows and columns on forms that include the Europe entity are displayed as read-only. Users can change data only for members to which they have write access.

Forms and Currencies

For a single-currency application, all entities use the currency selected when the application was created. For a multicurrency application, the selected Currency member on forms determines the currency in which values display. When the Currency member Local is selected for rows or columns, no currency conversion occurs for those rows or columns, and users can enter data for the entities in their native currency. If a currency member other than Local is selected, data values are converted to the selected currency for that row or column, and the form is read-only. You can enter data in rows or columns that have Currency or Local as the selected member. See [Designing Forms for Multiple Currencies](#).

Forms and Versions

For bottom-up versions, rows and columns with level 0 members allow data entry. Rows or columns set to a parent member are read-only. The point of view must also be set to the level 0 member to allow data entry on a bottom-up version. Target versions allow data entry in parent and children members.

Forms and Attributes

You can select members by selecting a shared attribute. For example, select the South attribute to include members having the South attribute. Values can be entered and saved into rows and columns that use attributes.

Forms and Shared Members

You cannot select shared members individually; instead, select them using a relationship function. For example, you could select an alternate functional rollup to include all members

under that rollup. Values can be entered into rows or columns that display shared members and are saved to the base members in the database. Shared members display the same as base members in forms.

Forms and Calculations

To optimize calculations, select row members using relationships (such as Descendants or Children) instead of selecting children individually. Calculating totals for the parent of individually selected children could take several passes, depending on the number of hierarchy levels.

Creating Simple Forms

Table 6-1 Simple Form Creation Checklist

Task	Want to Know More?
Set form layout, including: <ul style="list-style-type: none"> • Adding form rows and columns • Assigning dimensions to columns and rows • Selecting dimension members for users to work with • Setting grid properties for the form • Setting dimension properties • Adding formula rows and columns • Setting display properties for the form • Setting printing options for the form • Adding and updating validation rules in forms 	See Setting Form Layout .
Define page axis and point of view	See Defining Simple Form Page and Point of View .
Select members	See Using the Member Selector .
Set form precision, context menus associations, and whether to enable dynamic user variables	See Setting Form Precision and Other Options .
Select business rules and set properties	See Using Business Rules .
Define access permissions	See Setting Up Access Permissions .
Design formula rows and columns	See Designing Forms with Formula Rows and Columns .
Design data validation rules	See Designing Forms with Data Validation .

To create simple forms:

1. Select **Administration**, then **Manage**, and then **Forms and Ad Hoc Grids**.
2. Click **Actions**, and then select **Create simple form**.
3. On the **Properties** tab, provide a form name of up to 80 characters, and an optional description of up to 255 characters.
4. Select the **Plan Type** associated with the form. See [Forms and Plan Types](#).
5. **Optional**: Provide instructions for working with the form.
6. Click **Next** to specify the form layout. See [Setting Form Layout](#).

You can also create composite forms, which are forms that display several simple forms simultaneously. See [Creating Composite Forms](#).


Setting Form Layout

When you create forms, the **Layout** tab initially contains one row and one column, and all dimensions are in Point of View. When creating or editing forms, you can add rows and columns to a form, as necessary.

When setting row and column layout:

- Assign at least one dimension to the row and column axis.
- You cannot select the same dimension for multiple axes. (You can have dimensions on multiple axes if you set user variables in the point of view.)
- Select a dimension from any axis and drag it to the destination axis to move a dimension from one axis to another.
- Select display properties

To set or update the form layout:

1. Open the form, and then click **Layout**.
See [Selecting and Opening Forms and Folders](#).
2. **Optional:** Click  to select a dimension, and then drag it to **Rows** or **Columns**, or within a row or column.

Note:

Initially, all dimensions are in the form point of view. You can drag dimensions from point of view to rows, columns, or to page. You can also drag dimensions from any area in the grid (row, column, point of view, or page) to any other area.


3. **Optional:** Select another dimension, and then drag it to **Rows** or **Columns**, or within a row or column.
4. Select each dimension's members.
See [Using the Member Selector](#).
5. **Optional:** To rearrange the order of dimensions in rows or columns, click , and then select **Move Dimension Up** or **Move Dimension Down**.
6. Select a row header (such as 1 or 2) to set row properties, or a column header (such as A or B) to set column properties using the information in this table (options are listed on the right under **Segment Properties**).

Table 6-2 Segment Properties

Option	Description
Apply to all rows	Apply settings to all rows; available when there are two or more rows. Clear this option to set different properties for each row.
Apply to all columns	Apply settings to all columns; available when there are two or more columns. Clear this option to set different properties for each column.

Table 6-2 (Cont.) Segment Properties

Option	Description
Hide	Hides the column or row on the form
Read-only	Creates a read-only row or column, enabling comparison of old, read-only data with new, editable data
Show separator	Creates a bold border before the segment to visually distinguish it
Suppress hierarchy	Suppresses indentation.
Suppress missing data	Hides rows or columns without data. Clear to display rows or columns with "#MISSING" in cells when data is missing.
Column width	<ul style="list-style-type: none"> • Default: Use the column width defined at the grid level (under Grid Properties) • Small: Display seven decimal places. • Medium: Display 10 decimal places. • Large: Display 13 decimal places. • Size-to-Fit: Force all columns to fit in the displayed space based on the top data cell value. • Custom: Select a custom size to display more than 13 decimal places, up to 999 places.
Row height	<ul style="list-style-type: none"> • Default: Use the row height defined at the grid level (under Grid Properties). • Medium: Display standard row height. • Size-to-Fit: Force all rows to fit in the displayed space. • Custom: Select a custom size in pixels for the row height.

7. **Optional:** Add formula rows or columns. See [Adding Formula Rows and Columns](#).
8. **Optional:** Add or update data validation rules. See [Including Data Validation Rules in Forms](#).

Setting Form Grid Properties

Form grid properties set the general form row and column display.


To set form grid properties:

1. Open the form, and then click **Layout**.
See [Selecting and Opening Forms and Folders](#).
2. In **Grid Properties**, set general row and column properties using the information in this table:

Table 6-3 Form Grid Properties

Option	Description
Suppress missing blocks	<p>(Rows only) Improves the performance of the Suppress missing data setting when suppressing many rows, for example, 90% or more. The Suppress missing blocks setting can degrade performance if few or no rows are suppressed. Test forms before and after using this setting to determine whether performance is improved. Also test forms whenever you make significant changes to your application.</p> <p>With this setting selected, attributes may not display in forms, certain suppressed blocks may ignore Dynamic Calc members, and row members do not display as indented.</p>
Suppress missing data	Hides rows or columns without data. Clear to display rows or columns with "#MISSING" in cells when data is missing.
Suppress invalid data	Hides rows or columns with invalid data. Clear to display rows or columns that contain cells with data that is invalid. Cells with invalid data are read-only.
Default row height	<ul style="list-style-type: none">• Medium• Size-to-Fit: Force all rows to fit in the displayed space• Custom: Select a custom size in pixels for the row height
Default column width	<ul style="list-style-type: none">• Small: Display seven decimal places• Medium: Display 10 decimal places• Large: Display 13 decimal places• Size-to-Fit: Force all columns to fit in the displayed space based on the top data cell value.• Custom: Select a custom size to display more than 13 decimal places, up to 999 places

Table 6-3 (Cont.) Form Grid Properties

Option	Description
Enable Autosave	<p>Selecting this option has these effects in a simple form:</p> <ul style="list-style-type: none"> When planners move out of a cell, their changes are automatically saved, with no prompt or message. Cell values are aggregated to their parents, and the affected cells are displayed with a green background. Planners can successively undo actions with Ctrl+Z.
	<p> Note:</p> <p>For optimal Autosave performance, you must have only dense dimensions on rows and columns. However, if you must put a sparse dimension on either a row or a column, for improved Autosave performance on block storage databases, enable hybrid aggregation with the Oracle Essbase configuration setting <code>ASODYNAMICAGGINBSO</code>.</p>
Run Form Rules on Autosave	<p>If Enable Autosave is selected, this option becomes available. If Run Form Rules on Autosave is selected, dynamically calculated cells that depend on the changed and saved cell values (for example, a row that has a member formula that calculates a percentage of an aggregated parent value) are also updated and are displayed with a green background.</p>

- Click **Save** to save your work and continue, or click **Finish** to save your work and close the form.

Setting Dimension Properties

You can set and edit form dimension display properties, including whether to display the member name or alias in the form, hide the row or column, and permit users to view the member formula. These properties apply to row, column, page, and point of view dimensions.

To set dimension properties:

- Open the form, and then click **Layout**.
See [Selecting and Opening Forms and Folders](#).
- Click in a point of view, page, row, or column to set dimension properties.
- Select **Dimension Properties**:

Table 6-4 Dimension Properties

Property	Description
Apply to all row dimensions	Applies properties to all row dimensions
Apply to all column dimensions	Applies properties to all column dimensions
Apply to all page dimensions	Applies properties to all page dimensions
Apply to all POV dimensions	Applies properties to all point of view dimensions
Member Name	Displays the member name
Alias	Displays the member alias
Member Formula	Displays member formulas
Hide dimension	Hides the dimension
Show consolidation operators	Displays consolidation operators
Start expanded	Available only for dimensions on row or columns, choosing this option initially displays the dimension member list expanded.
Enable custom attributes	Available only for dimensions on row or columns, enables custom attributes.

- Click **Save** to save your work and continue, or click **Finish** to save your work and close the form.

Setting Display Properties

You can set and edit options for form display, such as hiding forms or displaying missing values as blank, in the **Layout** tab.

You can also enable account-level annotations. Users can add annotations to accounts in forms if they have write access to the account, entity, scenario, and version members. Account-level annotations can vary by different combinations of Scenario, Version, and Entity dimensions.

Notes:

- The Account dimension must be assigned to a row axis.
- Account, Entity, Versions, and Scenario dimensions cannot be assigned to the column axis.
- The Entity dimension can be assigned to the row, page, or Point of View axis.
- Version and Scenario dimensions must be assigned to the page or Point of View axis.

To set display properties:

- Open the form, and then click **Layout**.
See [Selecting and Opening Forms and Folders](#).
- Select **Display Properties**, and then select form options:

Table 6-5 Display Properties

Option	Description
Make form read-only	You cannot set this option for composite forms.

Table 6-5 (Cont.) Display Properties

Option	Description
Hide form	For example, hide forms that are part of composite forms or are accessed from menus or task lists.
Display missing values as blank	Leave form cells empty where data does not exist. If this option is not selected, empty cells display the text "#MISSING."
Enable account annotations	This option is available only if the Account dimension is on the row. For information about using account annotations, see the <i>Oracle Hyperion Planning User's Guide</i> .
Allow multiple currencies per entity	If the application supports multiple currencies, allow entities to support multiple currencies, regardless of base currency. Users can select currency for displayed cell values in forms.
Enable Mass Allocate	Users must have the Mass Allocate role to use this option. For information about using Mass Allocate, see the <i>Oracle Hyperion Planning User's Guide</i> .
Enable Grid Spread	For information about using Grid Spread, see <i>Oracle Hyperion Planning User's Guide</i> .
Enable cell-level document	(Default) Enable users to add, edit, and view documents in cells in the form, depending on access permissions. To prevent users from using documents in a form, clear this option. To use cell documents, see <i>Oracle Hyperion Planning User's Guide</i> , Setting Up Access Permissions .
Message for forms with no data	Enter text to display in form rows for queries without valid rows. Leave blank to display the default text, <i>There are no valid rows of data for this form</i> .

3. Click **Save** to save your work and continue, or click **Finish** to save your work and close the form.

Setting Printing Options

You can set and edit preferences for printing form information in the **Layout** tab.

To set printing options:

1. Open the form, and then click **Layout**.
See [Selecting and Opening Forms and Folders](#).
2. Select **Printing Options**, and then set preferences for printing form information:

Table 6-6 Printing Options

Option	Description
Include supporting detail	<p>Include supporting detail as extra rows in PDF files. Specify display format:</p> <p>Normal Order: Prints supporting detail in the same order as on the Supporting Detail page, after the member it is associated with</p> <p>Reverse Order: Prints supporting detail in reverse order, before the member associated with it. Supporting detail for children displays above parents, and the order of siblings is preserved</p>
Show comments	Display text notes associated with cells
Format data	Apply number format settings from the form to the displayed data
Show attribute members	If attribute members are selected in the form, display them in PDF files
Apply precision	Apply form precision settings (desired number of decimal points) to the displayed data in PDF files
Show currency codes	<p>If the form supports multiple currencies, display currency codes in the form and in PDF files. Whether currency codes display depends on whether currency codes are present on any member in the form.</p> <p>If a currency code is present on any member contained in the form, currency codes display in the form regardless of the selection for this check box. If currency codes are not present on members in the form, currency codes are not displayed.</p>
Show account annotations	If account annotations are enabled for the form, select to display account annotations in PDF files

You can also create reports for forms, as described in [Customizing Reports](#).

3. Click **Save** to save your work and continue, or click **Finish** to save your work and close the form.

Including Data Validation Rules in Forms

In the **Layout** tab, you can add and update validation rules to the grid, column, row, or cell. When rules are processed, they can change the color of cells, provide validation messages to users during data entry, and can change the promotional path for planning units. Validation rules are saved with the form.

Before adding data validation rules, it is important to consider the function the rule will perform, and to plan the rule scope. For detailed information and best practices for planning and implementing validation rules, see [Managing Data Validation](#).

To include data validation rules in forms:

1. Open the form, and then click **Layout**.
See [Selecting and Opening Forms and Folders](#).
2. Select **Validation Rules**, and then select an option:

 **Note:**

The menu options that display are context-sensitive, and depend on whether rules have already been added, and if you have previously selected a menu option. For example, if you right-click a cell that contains a rule and select **Copy Validation Rules**, the **Paste Validation Rules** menu option is displayed when you right-click another cell.

Table 6-7 Validation Rules Options

Option	Description
Add/Edit Validation Rules	Add or edit existing rules in the condition builder area of the Data Validation Rule Builder dialog box.
Copy Validation Rules	Copy the selected rules to be pasted to a new location.
Paste Validation Rules	Paste the previously copied rules to a new location.
Validate only for users with access to this form	If the currently logged in user does not have access to the form, do not execute validations associated with the form when validating the planning unit.
Validate only for pages with existing blocks	When enabled, Oracle Hyperion Planning intelligently figures out which page combinations have potential blocks and runs the validations only for those page combinations. There are a few exceptions to this. If a page combination has any Dynamic Calc, Dynamic Calc and Store, Label only, or Store with one child member, then that page is always loaded.
Validate only for cells and pages the user has access to	When enabled, validations are run as the currently logged in user and not as the administrator, which means the user's security will be applied to the form members.

3. Build and validate the rules as described in [Creating and Updating Data Validation Rules](#).
4. In the form, click **Next** to continue building the form, and then validate and save the form.

Setting Form Precision and Other Options

In **Other Options**, you set data precision, associate context menus with the form, and enable dynamic user variables.

You control data precision by applying minimum and maximum values for different account types. For example, you can truncate and round the decimal portion of longer numbers.

To set form precision and other options:








1. Open the form, and then click **Other Options**.
See [Selecting and Opening Forms and Folders](#).
2. In **Precision**, select options to set the number of decimal positions displayed in a cell for **Currency Values**, **Non-Currency Values**, and **Percentage Values**.


Specify **Minimum** values to add zeros to numbers with few decimal places. Specify **Maximum** values to truncate and round the decimal portion of longer numbers. For example:

Table 6-8 Data Precision Examples

Value	Minimum Precision	Maximum Precision	Displayed Value
100	0	Any	100
100	3	Any number greater than or equal to 3 or None	100.000
100.12345	Any number less than or equal to 5	None	100.12345
100.12345	7	None	100.1234500
100.12345	Any number less than or equal to 3	3	100.123
100.12345	0	0	100
100.12345	2	4	100.1234
100	2	4	100.00

Notes:

- By default, the precisions settings you select here override the precision set for the currency member (see [Creating Currencies](#)). If instead you want the currency member's precision setting to prevail for the form, select **Use Currency member precision setting**.
 - Precision settings affect only the display of values, not their stored values, which are more accurate. For example, if **Minimum Precision** is set to 2, and if Oracle Hyperion Planning spreads the value 100 from Q1 into the months Jan, Feb, and Mar, the month cells display 33.33 when they are not selected. When they are selected, they display their more accurate values (for example, 33.33333333333333). Because the number of decimal places for storing values is finite, when the values for Jan, Feb, and Mar aggregate back to Q1, 33.33333333333333 is multiplied by 3 and Q1's value displays 99.9999999999998 when you click into Q1.
3. In **Context Menus**, associate menus with the form by selecting them from **Available Menus** and moving them to **Selected Menus**:
-  moves selections
 -  moves all
 -  removes selections
 -  removes all
4. If you select multiple menus, click one of the following options to set the order in which they display:
-  moves the selection to the top of the order
 -  moves the selection up one spot in the order
 -  moves the selection down one spot in the order

-  moves the selection to the bottom of the order

Multiple menus display sequentially, with separators between them.

5. Select **Enable Dynamic User Variables** to allow dynamic user variables in the form (see *Oracle Hyperion Planning User's Guide*).
6. Click **Save**.



Creating Asymmetric Rows and Columns

Asymmetric rows and columns contain different sets of members selected across the same dimensions. For example:

Row/Column A: Scenario = Actual, Time Period = Q1

Row/Column B: Scenario = Budget, Time Period = Q2, Q3, Q4

To create asymmetric rows or columns:

1. Open the form, and then click **Layout**.
See [Selecting and Opening Forms and Folders](#).
2. Click  to select the dimension to modify.
3. Click  to the right of the dimension name, and then modify the members selected for this dimension. See [Using the Member Selector](#).
4. Click **Save** to save your work and continue, or click **Finish** to save your work and close the form.

Adding Formula Rows and Columns


Formula rows contain formulas that apply to form rows. Formula columns contain formulas that apply to form columns. For example, you can create a formula column (column D) that computes the percentage variance between the January sales (column A) and February sales (column B). The formula defined for a formula row or column applies to all row or column dimensions. To define or assign existing formulas to forms, select the appropriate row or column on the **Layout** tab and then display formula building options under **Segment Properties** (see [Creating Formulas](#)).

Tip:

Consider adding a formula row between two other rows to create a blank row. Blank rows are useful, for example, for visually separating subtotals and totals within a form.

To add formula rows and columns:


1. Open the form, and then click **Layout**.
See [Selecting and Opening Forms and Folders](#).
2. On the **Layout** tab, right-click **Rows** or **Columns**.
3. Select **Add Formula Row** or **Add Formula Column**.

4. Click the new **Formula Label** that is displayed in the row or column, and then enter the formula name.
5. Click the row or column number and specify any of the following displayed in the **Segment Properties** pane to the right:
 - **Hide** hides the row or column
 - **Show separator** displays the row or column separator
 - **Display formula on form** displays the formula on the form when you click  in the row or column header.
6. For each dimension in **Formula Data Type** in the right pane, select a data type for the formula result.

The data types are:

 - Currency
 - Non Currency
 - Percentage
 - SmartList

If you select SmartList, select a Smart List from the drop-down list next to the data type.



 - Date
 - Text
7. Define the formula to use for the row or column by entering the formula name in the **Formula** field, and then clicking . See [Editing Formulas](#).
8. Click **Validate** to ensure that the formula does not contain any errors.
9. Click **OK** to save the formula and to close the **Formula** window.

Defining Simple Form Page and Point of View



You can select dimensions and members for the page axis and point of view. Point of view dimensions and members must be valid for the form plan type and not assigned to a page, column, or row axis. The point of view sets the unique dimension members that define intersections of data.

When you set user variables for forms, the variable name displays in point of view. See [Managing User Variables](#).

To define page axis and point of view:

1. Open the form, and then click **Layout**.
See [Selecting and Opening Forms and Folders](#).
2. Click  and then drag that dimension to **Page** to add it to the form page axis.
3. Click  for each page axis dimension and select members. See [Using the Member Selector](#).

You can specify the number of members in a page dimension that enables a search option. Select **File**, then **Preferences**. In **Display Options**, type a value for **Allow Search When Number of Pages Exceeds**.

- Repeat Steps 2 and 3 to assign multiple dimensions to the page axis.
Assigning multiple dimensions to the page axis enables planners to select dimensionality while entering data. Users can select **Display Options** to specify whether Oracle Hyperion Planning sets the page selection to the most recently used selection.
- In **Dimension Properties**, select or clear options for page dimensions. See [Setting Dimension Properties](#).
- Optional:** Click  and then drag that dimension to **Point of View** to add it to the form point of view. Repeat this action for each dimension you want to move to **Point of View**.
- In **Point of View**, click  for each dimension and then select members.
You can also create members that do not yet exist "on the fly".
See [Using the Member Selector](#).
- In **Dimension Properties**, select or clear options for point of view dimensions. See [Setting Dimension Properties](#).
- Click **Save** to save your work and continue, or click **Finish** to save your work and close the form.

Importing Form Definitions

Use the `ImportFormDefinition` utility to import a form definition, but not data, from a text-based file into a Oracle Hyperion Planning form. You can import rows, columns, or both. Planning imports only rows or columns that contain data. You must run the utility on a Windows system.

To import form definitions:

- Prepare the form for importing.
See [Preparing Forms](#).
- Prepare the data file.
See [Preparing Data Files](#).
- Run the utility.
See [Importing Form Definitions](#) and [ImportFormDefinition Examples](#).

Preparing Forms

Before importing rows and columns for form definitions, create the form by setting up dimensions on the row, column, page and Point of View, depending on your requirements. Typically you define column layout when defining forms and use `ImportFormDefinition` to import only rows. The resulting form looks like the form definition.

The rows imported from the data file are based on members specified on the form and import options, determining which data is imported. See [ImportFormDefinition Examples](#).

Preparing Data Files

`ImportFormDefinition` imports data from a text-based file in Oracle Essbase Column Export Format. You can generate the file directly, or, if you have a method for loading data into Essbase, you can load the data and generate the file.

To create an Essbase Column Export Format file using Essbase, use Administration Services Console, and select Database, then Export. Specify the Server File Name and select Export in Column Format. (Oracle recommends that you also select Level 0 Data.) You need not modify the file after exporting it from Essbase.

If you generate the file yourself:

- The file's first line represents the column of the data file. It must be a list of members from one dimension.
- Each line after the first line must include a member from every dimension other than the one representing the column, followed by data.
- Member names must be enclosed in double quotation marks.
- The delimiter must be a space.
- Data must not be enclosed in double quotation marks.
- Data cells that are blank must include "#MISSING."

The layout of the form, not the format of the data file, determines how the resulting form displays. You can use the same data file to load different forms.

Importing Form Definitions

The `ImportFormDefinition` utility is in the `planning1` directory. For the full path to `planning1`, see [About EPM Oracle Instance](#).

To launch `ImportFormDefinition`:

1. From the `planning1` directory, enter the command using this syntax:

```
ImportFormDefinition [-f:passwordFile] /A:appname /U:username /F:formname /  
D:filename [/AR] [/AC] [/SR] [/SC] [/KC] [/KR]
```

For a list of parameters, see [ImportFormDefinition Utility Parameters](#).

2. If prompted, enter your password.

For example:

```
ImportFormDefinition /A:MyPlan /U:Admin /F: "My Budget" /D:exportfilename /AR /-  
AC
```

Creating the form can take some time, depending on the amount of data in the file.

`ImportFormDefinition` imports the definition to the Oracle Hyperion Planning form, ensuring that all cells having data in the data file are represented on the form. For cells in the data file that contain "#MISSING," rows or columns are not added to the form.

Notes:

- If you disable importing rows or columns, `ImportFormDefinition` filters imported data by rows or columns defined on the form.
- If you run `ImportFormDefinition` more than once, it merges new results with the existing form definition, and, if you also specify sorting, sorts new and existing rows or columns by dimension order.

ImportFormDefinition Utility Parameters

Table 6-9 ImportFormDefinition Utility Parameters

Setting	Purpose	Required?
<code>[-f:passwordFile]</code>	If an encrypted password file is set up, you can use this option as the first parameter in the command line to run the utility with the full file path and name specified in <i>passwordFile</i> . See Suppressing Password Prompts in Planning Utilities .	No
<code>/A</code>	Application name.	Yes
<code>/U</code>	Administrator user name.	Yes
<code>/F</code>	Form name.	Yes
<code>/D</code>	Name and location of the Oracle Essbase Column Export Format data file. The location can be the full path and file name, or any format required for the operating system to find the file.	Yes
<code>/AR</code>	Add rows from the data file (on by default). Disable by specifying <code>/-AR</code> . For example, you can define rows in Oracle Hyperion Planning and import only column definitions.	No
<code>/AC</code>	Add columns from the data file (on by default). Disable by specifying <code>/-AC</code> .	No
<code>/KC</code>	Keep member selections for columns in the form (on by default). Clear columns by specifying <code>/-KC</code> . Forms must have at least one column definition. If you clear columns but do not add them, the form is not saved, and an error displays.	No
<code>/KR</code>	Keep member selections for rows in the form (on by default). Clear member selections in rows by specifying <code>/-KR</code> . Forms must have at least one row definition. If you clear rows but do not add them, the form is not saved, and an error displays.	No
<code>/SR</code>	Sort rows in the form (on by default). Disable by specifying <code>/-SR</code> .	No
<code>/SC</code>	Sort columns in the form (on by default). Disable by specifying <code>/-SC</code> .	No

ImportFormDefinition Examples

Define members on each axis of the form appropriately and define import options correctly because this affects which data is imported.

To import only rows that contain data for the specified columns:

1. In Oracle Hyperion Planning, specify the form columns (for example, Descendants Inclusive of YearTotal).
2. For the dimension for which to import members to the row, add the dimension root to the form design.

For example, to load accounts on the row, place the Account dimension root on the form's row.

3. When you run `ImportFormDefinition`, use these options: `/AR /-AC`

Accounts are loaded from the data file if members from each dimension making up a cell match members on the form columns, pages, and Point of View. The added rows are filtered by members on the page. For example, if you place some members on the page, only accounts that contain data for those members are added to the form. Rows are filtered by the members on the Point of View. If the data file contains data for the salary account for 2008, but only 2009 is on the Point of View, the salary account is not added to the row although it exists in the data file.

Creating Composite Forms

Composite forms display several forms simultaneously, even those associated with different plan types. Users can enter data and see results aggregated to an upper-level intersection, such as Total Revenue.

To create composite forms:

1. Select **Administration**, then **Manage**, and then **Forms and Ad Hoc Grids**.
2. Select the folder in which to store the form. See [Selecting and Opening Forms and Folders](#).
3. Click **Create Composite Form** above the list of forms.
4. In the **Properties** tab, enter a form name of up to 80 characters, and an optional description of up to 255 characters.
5. **Optional:** Select **Hide Form** to hide the form.
6. **Optional:** Enter instructions for the form.
7. Set the composite form layout. See [Setting Composite Form Layout](#).
8. Set the composite form section properties. See [Setting Composite Form Section Properties](#).
9. Set the composite form Point of View and Page display options. See [Setting Composite Form Point of View and Page Dimensions](#)
10. Click **Save** to save your work and continue, or click **Finish** to save your work and close the form.

Setting Composite Form Layout

Oracle Hyperion Planning provides tools that allow you to create whatever composite form layout is best for your application. Each area in the composite form is called a section. Initially, you specify whether to divide the composite form layout into two side-by-side sections, or two sections that are stacked one above the other. There is also a custom layout option.

To set composite form layout:

1. Open the composite form, and then click **Layout**.

See [Selecting and Opening Forms and Folders](#).

2. In **Select Layout**, select an option:



- **Custom Layout** to create your own composite form layout.

 **Note:**

The custom layout option is selected by default.



- **2-Row Layout** to split the composite form into two sections, one on top of the other, divided by a horizontal line.




- **2-Column Layout** to split the composite form into two side-by-side sections divided by a vertical line.

The selected layout is displayed.

3. Add, rearrange, or delete simple forms as desired.

See [Adding Simple Forms to a Composite Form Layout](#), [Rearranging Forms in the Layout](#), and [Deleting Simple Forms from a Composite Form](#).

4. **Optional:** Click  in the upper right side of a section to select the following additional layout options for that section:

- **Split Horizontally** to split the section into two sections, one above the other.
- **Split Vertically** to split the section into two side-by-side sections.

 **Note:**

When you split a composite form section that contains simple forms, the simple forms remain in the original section. For example, if you split a section vertically, the original section is divided into two side-by-side sections. The simple forms from the split section are included in the left section, and the right section is empty.

- **Delete** to remove a section from the composite form.

When you delete a section from a composite form, the simple forms included in that section are also deleted from the composite form, unless they are included in other sections of the composite form.

- **Add Form** to display the Form Selector dialog box where you can select additional forms to add to the layout.
- **Group as Tabs** to display the forms in that section as tabs.
- **Ungroup Tabs** to clear **Group as Tabs**.



5. Click **Save** to save the Composite Form layout.

 **Tip:**

You can edit a simple form from within a composite form. Right-click the simple form, and then select **Form Designer**. Edit the form as described in [Editing Forms](#). Access permissions apply as described in [Forms and Access Permissions](#).

Adding Simple Forms to a Composite Form Layout

To add a simple form to a section in a composite form, do one of the following:

- Drag a form from the **Forms in <Form Folder>** pane to the desired section.
- Click in the desired section, select , and select **Add Form**. In the Form Selector dialog box, select a form and click **OK**.
- Expand Section Properties and click . In the Form Selector dialog box, select a form and click **OK**.

When you are adding simple forms to a composite form, note the following:

- Composite forms can contain simple forms and ad-hoc forms.
- During runtime, the simple forms selected for the composite form display from left to right, and then from top to bottom within each composite form section.
- If you select Group as Tabs, the form displays in the order selected.
- You can drag simple forms between sections of a composite form.

Rearranging Forms in the Layout

To rearrange the simple forms in the composite form layout, expand Section Properties, select a form, and click an arrow key. You can:

- Move the form to the top
- Move the form up
- Move the form down
- Move the form to the bottom

Editing Simple Forms from within a Composite Form


While editing a composite form, you can edit a single form from the **Layout** tab. This option is not available for ad hoc forms. Access permissions apply as described in [Forms and Access Permissions](#).

To edit a simple form from a composite form:

1. Within the composite form, click the **Layout** tab.
2. Right-click a simple form, and then select **Form Designer**.
3. Edit the simple form as described in [Editing Forms](#).

Deleting Simple Forms from a Composite Form

To delete a simple form from a composite form, do one of the following:

- Right-click the form and select **Delete**.
- Select the form in **Section Properties**, and click .
- Uncheck the form in the Form Selector dialog box and click **OK**.

Setting Composite Form Section Properties

Each section in a composite form is associated with properties set during creation. You can edit these properties after creating a composite form.

To set composite form properties:

1. Open the composite form, and then click **Layout**.
See [Selecting and Opening Forms and Folders](#).
2. Expand **Section Properties**.
3. Click in a composite form section and set the properties as desired.

Table 6-10 Composite Form Section Properties




Option	Description
Forms	Displays the simple forms in the section. The following options are available for each form selected: <ul style="list-style-type: none"> • Display forms as tabs • Add form • Remove form • Edit form label • Move to top • Move up • Move down • Move to bottom
Name	Section name to be displayed at the top of the section in Preview mode and at runtime. Select  to select a text style and color for the section name.
Height	Section height. Select: <ul style="list-style-type: none"> • Automatic to have Oracle Hyperion Planning set the height. • % (percentage sign) to set section height to a percentage of the composite form height.
Width	Width of the section. Select: <ul style="list-style-type: none"> • Automatic to have Planning set the width. • % (percentage sign) to set section width to a percentage of the composite form width.

Table 6-10 (Cont.) Composite Form Section Properties

Option	Description
Forms per Row	<p>Select:</p> <ul style="list-style-type: none"> • Automatic to have Planning set the number. • Select a number from 1 to 20. <p>The default is one form per row. If Forms per Column is set to a value other than Automatic, Forms per Row is set to Automatic.</p> <div style="border: 1px solid #0070C0; padding: 10px; margin-top: 10px;"> <p> Note:</p> <p>If you have grouped the forms as tabs, this option is not available.</p> </div>
Forms per Column	<p>Select:</p> <ul style="list-style-type: none"> • Automatic to have Planning set the number. • Select a number from 1 to 20. <p>The default is one form per column. If Forms per Row is set to a value other than Automatic, Forms per Column is set to Automatic.</p> <div style="border: 1px solid #0070C0; padding: 10px; margin-top: 10px;"> <p> Note:</p> <p>If you have grouped the forms as tabs, this option is not available.</p> </div>
Set scope for all common dimensions as global	<p>Sets all the common dimensions across all the sections in the composite form to global and displays a list of the global dimensions in Page and Point of View in the Global Dimensions properties.</p>

Setting Composite Form Point of View and Page Dimensions

The composite form point of view and page dimensions specify where within a composite form each Point of View and Page dimension name displays. When you select a section in a composite form, the right panel displays:

- **Global Layout Dimensions**, which list the Point of View and Page dimensions that display in the composite form heading.

Only dimensions that are common to all simple forms in all sections of the composite form and that contain the same members can be designated as Global.

- **Common Dimensions**, which list the Point of View and Page dimensions common to all the simple forms included in the selected composite form section.

You can specify where common dimensions display in composite forms. Common dimension display choices are:

- **Local** displays the dimension name in the simple form heading.
- **Section** displays the section name in the section heading.

Only dimensions that are common to all simple forms in a section and that contain the same members can be displayed in the section heading.

- **Global** displays the dimension name in the composite form heading.

Creating Master Composite Forms

You can design composite forms that have one master form and multiple simple forms. When you do so, the selection of members in the master form automatically filters to the members in the simple forms, and the simple forms show only the details that are relevant to the members highlighted in the master form.

For example, assume that a user is looking at a new computer line item in a form and wants to see the cash flow impact from this line item. In this scenario, you could design a composite form that includes the following forms:

- A master form called "New Computers" that contains the following dimensions and members:
 - Entity: MA
 - Scenario: Plan
 - Version: Working
 - Currency: Local
 - Year: No Year
 - Period: Beginning Balance
 - Asset Class: Computers
 - Line Item: Base SP1
- A simple form called "Cash Flow Impact".

In the master composite form, the user highlights the row `Computers/Base SP1`.

Figure 6-1 Master Composite Form: "New Computers"

MA Plan Working Local Beginning Balance No Year Refresh												
C16												
	A	B	C	D	E	F	G	H	I	J	K	
			Asset Description	Asset Units	Asset Rate	Basic Cost	Salvage	Purchase Date	In Service Date	Premature End Date	Reason ended	
2	Furniture and Fixtures	Base SP1	Office Furniture	0	1,000	1,000	0	1/2/2009	1/2/2009	12/10/2009	Transfer Out	
3		Base SP2	Office Desktops	0	100	100	5	1/2/2009	1/2/2009		None	
4	Computers	Base SP1	Laptops	0	50	250	0	1/2/2009	1/2/2009		None	
5	Tangible Assets	Total Specified				1,250					None	

The simple form, "Cash Flow Impact" is filtered to show only the data that is relevant for the members highlighted in the master composite form, "New Computers": Computers, Base SP1, Plan, Working, and MA.

Figure 6-2 Simple Form: "Cash Flow Impact"

MA		Computers		Plan	Working	Base SP1	FY10	Local	Refresh										
F16		fx																	
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P			
1		-YearTotal	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec					
2	Cash Outflow from Capital Additions	250	250																
3	-Net Cash Flows	-250	-250																
4																			

To designate a form as a master composite form:

1. Open the composite form, and then click **Layout**.
See [Selecting and Opening Forms and Folders](#).
2. Right-click the form, and then select **Tag as Master Composite Form**.



indicates that the form is a master composite form.

Note:

The master composite form applies to the entire composite form. So, for a composite form, there can be only one master form across all its sections.

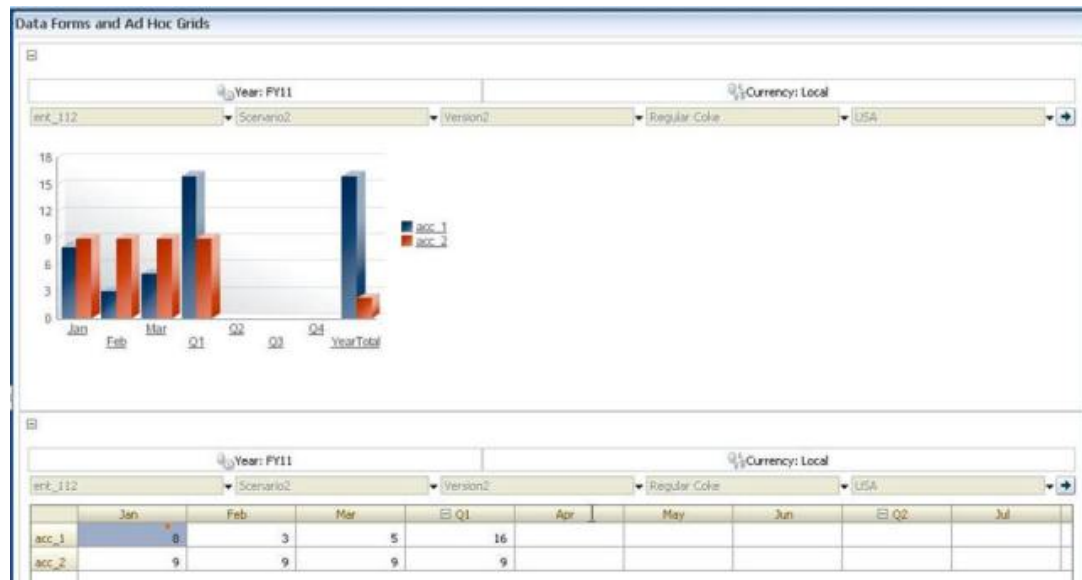
To filter the data in a simple form (or forms) that is relevant to the data in a master composite form, right-click the master composite form and select **Apply Context**.

Embedding Charts in Composite Forms

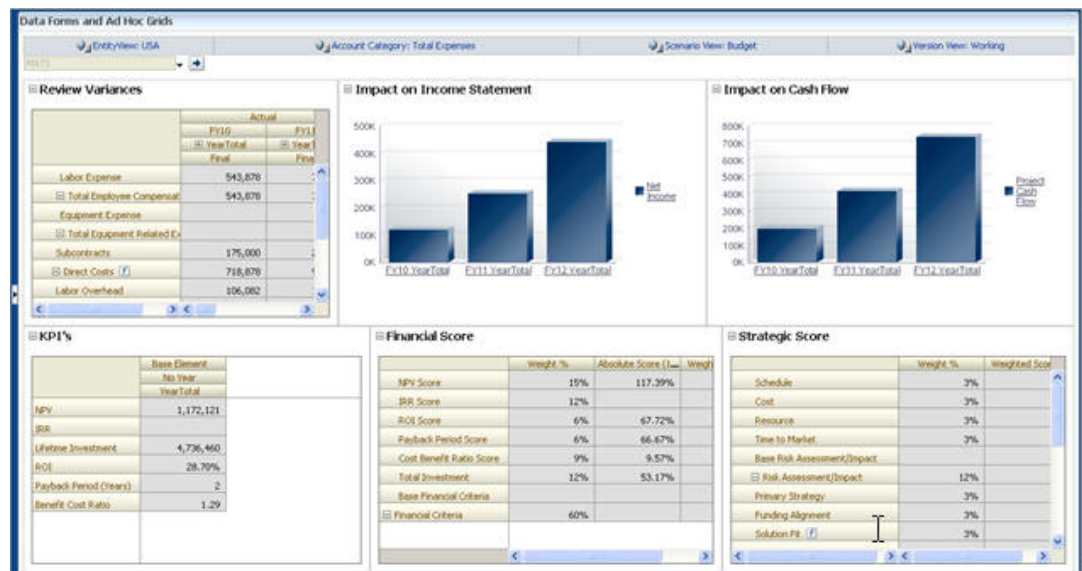
Administrators can design composite forms to display the data in sections as charts. Planners can also drill down to the next level by clicking the underlined links or chart areas.

Design suggestions:

- Display the top section as a chart and the bottom section as a grid, so that planners can see the effect of data they enter in the bottom grid (when saved) as a chart on the top.



- Include the same ad hoc grid twice, one to display as a grid and the other to display as a chart. Users can then perform ad hoc operations (such as **Zoom In**, **Pivot To**, and **Keep Only**) on the grid and view the changes in the chart.
- Create dashboards. For example:



To embed charts in composite forms:

1. Create or edit the composite form, and then click **Layout**.
See [Selecting and Opening Forms and Folders](#).
2. Click a composite form section, and then right-click on a form.
3. Select **Display as Chart**.
Display as Chart toggles with **Display as Grid**, allowing you to switch between them.
4. On **Chart Properties**, select a chart type:

Table 6-11 Chart Types

Chart Type	Description
Bar	The length of each bar proportionally represents a value over an independent variable (for example, time).
Horizontal Bar	Like the regular bar chart, but turned on its side so that the dependent variable is displayed on the horizontal axis.
Line	Displays data points (for example, sales of various product lines) over time, connected by lines.
Area	Like the Line chart, but the area between the axis and the line is emphasized with color.
Pie	Each slice of the pie chart proportionally represents a class of data in relation to the whole.
Scatter	Each point represents the distribution of data for two variables.

Read the onscreen text for advice on selecting a chart type.

5. Click **OK**.
6. **Optional:** To set where the chart displays the values that the chart represents (called the Legend), click **Options**, then click **Legend**, select one of the following, and then click **OK**:
 - **Right:** To display the legend to the right of the chart (the default).
 - **Bottom:** To display the legend at the bottom of the chart.
 - **Left:** To display the legend to the left of the chart.
 - **Top:** To display the legend at the top of the chart.
7. **Optional:** To set where the chart labels (that is, the member names or aliases) are displayed, on **Options**, click **Label**, select one of the following, and then click **OK**.
 - **Outside Max:** To display the label above bar charts or, for non-bar charts, display the label above the data point for positive values and below the data point for negative values. **Outside Max** is the default.
 - **Center:** To display the label centered on bar charts, or for non-bar charts, display the label above the data point for positive values and below the data point for negative values.
 - **Inside Max:** To display the label on the bar, near the top, or for non-bar charts, display the label below the data point for positive numbers and above the data point for negative numbers.
 - **Inside Min:** To display the label inside on the bar, near the bottom, or for non-bar charts, display the label above the data point for positive values and below the data point for negative values.
 - **Max Edge:** To display the label on the bar, or for non-bar charts, display the label at the data point.

Designing Specific Types of Forms

Related Topics

- [Designing Forms for Multiple Currencies](#)
- [Designing Forms for Drill-Through Information](#)
- [Designing Forms with Formula Rows and Columns](#)
- [Designing Forms with Data Validation](#)
- [Designing Forms with Global Assumptions](#)
- [Designing Forms for Rolling Forecasts](#)

Designing Forms for Multiple Currencies

To allow users to work with currencies other than entities' base currencies, perform one task:

- Select members from at least two currencies to compare converted currencies in the same form.
- Assign the Currency dimension to the page axis and select reporting currencies as members to convert currencies for all the members in the form. Users can then select a currency member from the page axis and launch the Calculate Currencies business rule to view values in that currency.

Designing Forms for Drill-Through Information

If forms contain members whose data is loaded from a source such as Oracle Hyperion Financial Data Quality Management, Enterprise Edition, users can drill through to view more details for the cell data source. To enable forms for drill through, complete these tasks when designing forms:

- Within FDMEE, complete setup tasks, and load data or metadata to Oracle Hyperion Planning. See the *Oracle Hyperion Financial Data Quality Management, Enterprise Edition Administrator's Guide*.
- Give appropriate access permissions, for example, to the form and members, as described in [Setting Up Access Permissions](#). Drill-through is enabled against all dimensions that are sourced from FDMEE. If drill-through is enabled in a cell to which a user has write access, the drill-through icon continues to display when users update the cell. However, the update does not tie back to the source data when the user drills through.
- For a multicurrency application, all currencies for an entity in the source system can be loaded. Exchange rates are loaded into the exchange rate table in Planning, and currency conversion is completed within Planning.

When users print forms that include cells with drill-through information, a drill-through icon is displayed in those cells.

Designing Forms with Formula Rows and Columns

Formula rows and columns contain formulas that perform mathematical calculations on grid members. For example, you might want to rank values in a particular column or calculate variance between two rows. A formula consists of grid references, arithmetic operators, and mathematical functions. To define or assign existing formulas to forms, select the appropriate row or column on the **Layout** tab and then make selections under **Segment Properties**.

See [Adding Formula Rows and Columns](#). For information about creating formulas and using the mathematical functions, see [Form Formula Functions](#).

Designing Forms with Data Validation

You can design forms that include predefined data validation rules that help implement business policies and practices. You can specify cell colors and data validation messages that are generated on the form if entered data violates a validation rule. Data validation rules are saved as part of the form. See [Including Data Validation Rules in Forms](#) and [Managing Data Validation](#).

Designing Forms with Global Assumptions

Form designers typically create a specific-purpose form that stores global assumptions (also called "drivers" or "driver data"), such as a depreciation value, tax rate, or unit price. This is useful if you want to migrate the form's driver data using Oracle Hyperion Enterprise Performance Management System Lifecycle Management, for example, from a test environment to a production environment. Such forms are typically small, and are used as "lookup tables" that drive calculations.

To enable transferring global assumptions from a test to production environment, during form design for a simple form, in the **Layout** tab select **Grid Properties** and **Global Assumptions Form**. Then update the form to store driver data such as a tax rate. When migrating with Lifecycle Management, this enables migrating the driver data contained in the tagged form. Note that global assumptions cannot be used with composite forms or forms with multiple segments.

Designing Forms for Rolling Forecasts

About Rolling Forecasts

In a traditional forecast, the forecast cycle is always tied to the fiscal year end, and the months in the forecast period keep reducing as the months in the fiscal year progress.

Rolling forecasts differ from traditional forecasts in that they are continuous without regard to the annual fiscal year end period. The periods in a rolling forecast roll along based on the predefined window for the rolling forecast. The periods are generally defined on a monthly or quarterly basis. Monthly rolling forecasts are generally in 12-month, 18-month, or 24-month cycles. In a 12-month cycle, the 12-month period constantly shifts each month, and every month the forecast is for the next twelve months without regard to the actual fiscal year end.

For example, assume a company has a fiscal calendar for July through June. In the first month of year (Jul, FY11) the company's planners fill in the forecast scenario for the periods Jul 11 – Jun 12. In the next month (Aug 11), the planners again fill in the forecast scenario with numbers for the next 12 months (Aug 11- Jul 12), even though the period of Jul 12 pertains to the next fiscal year of Jul FY12-Jun FY13.

Following are some examples of rolling forecasts:

Figure 6-3 12-Monthly Rolling Forecast

	Year and Period in Columns With No Additional Segment												FY13	FY13	Total		
	FY12 Jul	FY12 Aug	FY12 Sep	FY12 Oct	FY12 Nov	FY12 Dec	FY12 Jan	FY12 Feb	FY12 Mar	FY12 Apr	FY12 May	FY12 Jun					
Actual	50	50	50														
Plan/Budget	200	200	200	200	200	200	200	200	200	200	200	200	300	300	300		
12 month Rolling Aug	50	75	75	75	75	75	75	75	75	75	75	75	125				
12 month Rolling Sep	50	50	75	75	75	75	75	75	75	75	75	75	125	125			
12 month Rolling Oct	50	50	50	75	75	75	75	75	75	75	75	75	125	125	125		

Figure 6-4 Quarterly Rolling Forecast

	FY12 Q2	FY12 Q3	FY12 Q4	FY13 Q1	FY13 Q2	FY13 Q3	FY13 Q4
FY12 Q2 Review	F	F	F	F			
FY12 Q3 Review	A	F	F	F	F		
FY 12 Q4 Review	A	A	F	F	F	F	
FY13 Q1 Review	A	A	A	F	F	F	F

Figure 6-5 Quarterly Trailing Rolling Forecast (Rolling Quarters with a Cumulative Total)

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	12 qtrs rolling
Project 1	25	35	45	55	65	75	85	95	105	115	125	135	960
Project 2	50	60	70	80	90	100	110	120	130	140	150	160	1260
Project 3	75	85	95	105	115	125	135	145	155	165	175	185	1560
Project 4	100	110	120	130	140	150	160	170	180	190	200	210	1860
Project 5	125	135	145	155	165	175	185	195	205	215	225	235	2160

Figure 6-6 Rolling Forecast Where There are Additional Segments for Actual and Plan Year

	Year and Period in Columns												12 month Rolling Aug			Actual	Plan
	FY12 Jul	FY12 Aug	FY12 Sep	FY12 Oct	FY12 Nov	FY12 Dec	FY12 Jan	FY12 Feb	FY12 Mar	FY12 Apr	FY12 May	FY12 Jun	FY13 Jul	FY13 Aug	FY13 Sep	FY12 YearTotal	FY13 YearTotal
Account 1	50	75	75	75	75	75	75	75	75	75	75	75	125			600	3600

Creating Rolling Forecasts

 **Note:**

Only administrators can create and work with rolling forecasts. This includes the ability to see the rolling forecast option when designing a form, the ability to move a rolling forecast from the user interface, and the ability to delete or modify substitution variables.

To design a form for a rolling forecast:

1. Create a new form.

See [Creating Simple Forms](#).



2. In the Layout tab, drop the **Year** and **Period** dimensions in the column axis.

3. Right click the column segment header and select **Rolling Forecast Setup**.

The Rolling Forecast Setup menu option is only available when the Year and Period are in the same grid axis (row or column).

4. In the **Rolling Forecast Setup** dialog box, enter the following information:

Table 6-12 Rolling Forecast Setup Options

Option	Description
Prefix	Allows rolling forecast substitution variables to be distinct from other substitution variables; for example, 4QRF designates that the forecast is a 4-quarter rolling forecast.
Reuse existing substitution variables	Select if you wish to specify a prefix that you have previously used.
Start Year	The year in which the rolling forecast starts; for example, FY11. Either enter the start year or click  to open the Member Selection dialog box. If you entered a prefix that matches the prefix of an existing rolling forecast substitution variable and selected Reuse existing substitution variables , the Start Year is automatically filled in with the start year of the existing substitution variable.
Start Period	The period in the year in which the rolling forecast starts; for example, Q1. Either enter the start period or click  to open the Member Selection dialog box. If you entered a prefix that matches the prefix of an existing rolling forecast substitution variable and selected Reuse existing substitution variables , the Start Period is automatically filled in with the start period of the existing substitution variable.
Number of Periods	Number of year/period combinations that will be generated as separate segments.

5. Click **Generate**.

The defined substitution variables are created, and additional column segments are created in the form that contain the substitution variable combinations for the rolling forecast.

Notes:

- Substitution variables are based on the periods selected for the Start Year and Start Period. See [About Selecting Substitution Variables as Members](#).

- When designing a form, if the Rolling Forecast Setup dialog box is brought up from a row or column with the Year and Level 0 time period selected (for example, FY12/Jan), the Start Year and Start Period are automatically filled in. Values are not automatically filled in if members in the column are selected using functions, variables, or non-level-0 members.
- To reuse the rolling forecast variables in a different form, right-click the column header in the new form to invoke the Member Selector.

Modifying Rolling Forecast Variables

Administrators can revise the values for rolling forecast substitution variables directly in the form.

To modify rolling forecast variables in a form:

1. Select **Administration**, then **Manage**, and then **Forms and Ad Hoc Grids**.
2. Open the rolling forecast form.
3. Right-click any column in the form and select **Set Rolling Forecast Variables**.
4. In the **Set Rolling Forecast Variables** dialog box, enter or edit values for the Year and Period dimensions.

You can shift the values up or down by changing the selection next to **Shift Values By**. When you change the selection next to **Shift Values By**, the values for the Year and Period dimensions are automatically repopulated to show the resulting year and period values after the shift.

5. Click **Apply**.

The new values flow through to all the forms where these substitution variables are used, and those forms will reflect the changes.

Working with Forms and Form Components

Selecting and Opening Forms and Folders

Use these procedures to select and open form folders and the forms they contain. For ease of editing, administrators can open non-ad hoc forms for editing directly from the end-user interface.

To select and open non-ad hoc forms from within the end-user interface:





1. Open the data form.
2. Click the **Form Designer** icon  at the top of the page.

The form opens in edit mode in a new tab.

To select and open forms or form folders from within the administrator's interface:

1. Select **Administration**, then **Manage**, and then **Forms and Ad Hoc Grids**.
2. Perform one of the following steps:
 - To open a form folder, select a form folder beneath **Form Folders**.
 - To open a form, select a form from the list displayed beneath **Form** when the appropriate form folder is open.

After you select a form folder, use the buttons next to **Form Folders** to create, rename, and assign access to the folder. After displaying a form, use the buttons above **Form** to create, edit, move, delete, and assign access to forms. The icon next to the form name indicates the type of form:

-  Simple form
-  Composite form
-  Master composite form
-  Ad hoc grid

For information about setting up ad hoc grids, see *Oracle Hyperion Planning User's Guide*.

Previewing Forms

While you are designing forms, you can preview the dimensions that are assigned to the Point of View, columns, rows, and page axes. Previewing displays member attributes, alias, and data associated with forms, although new data cannot be entered.

Previewing completes regular form design validation checks, and checks for proper evaluation of any data validation rules included in the form. Data validation rules must be properly completed before the form can be saved. In addition, data validation rules are saved as part of the form. If you do not save changes to a form, any data validation rule changes made after the form was last saved are lost.

To preview a form's design:



1. With a form open, click **Preview**.
The form opens in edit mode in a new tab.
2. Resolve any issues reported during the design validation checks, including any issues with data validation rules.
3. Save the form to ensure that updates are saved, including any changes to data validation rules.



Printing Form Definitions

Administrators can print form definition reports that include information on dimension members, business rules, access permissions, and other form components.

You can also create reports for form definitions, as described in [Customizing Reports](#).

To create and print form definition reports:

1. Select **Tools**, and then **Reports**.
2. Select **Forms**.
3. Select the form definitions to print by moving them from **Available Forms** to **Selected Forms**:
 - To add one or more selected forms, click .
 - To add all forms, click .



- To remove one or more forms, click .
 - To remove all forms, click .
4. **Optional:** Select **Include Member Selection List** to include column and row members on the report.
 5. **Optional:** Select **Include Business Rules** to include associated business rules.
 6. Click **Create Report**.
Adobe Acrobat generates a consolidated report, including:
 - Plan type
 - Description
 - Column dimension and members and additional column definitions
 - Row dimension and members and additional row definitions
 - Page and Point of View dimensions
 - Form access permissions
 - Associated business rules
 7. To print the report, select **File**, and then **Print** on the Adobe toolbar.

 **Note:**

To ensure that multibyte characters display in reports, see "Multibyte Characters" in the *Oracle Enterprise Performance Management System Installation and Configuration Troubleshooting Guide*. To see the correct language glyph reflected in the reports, you must also make the fonts available in the `java.home` directory. The Oracle Hyperion Planning directory points to the location `EPM_ORACLE_HOME\common\JRE\Sun\1.6.0\lib\fonts`.

Searching for Forms

To search for forms:

1. Select **Administration**, then **Manage**, then **Forms and Ad Hoc Grids**.
2. For **Search**, enter part or all the form name.
Ignoring capitalization, Search finds the next match.
3. Click  to search forward (down) or  to search backwards (up).

Editing Forms

You can edit the layout, members, and properties of both simple and composite forms. For example, you can add formula rows or columns to a simple form, or add forms to a composite form.


Editing Simple Forms

To edit simple forms:

1. Select the form, then click **Edit** (see [Selecting and Opening Forms and Folders](#)).
2. Select:
 - a. **Properties** to edit the form name, description and instructions. See [Creating Simple Forms](#).
 - b. **Layout** to edit form layout. See [Setting Form Layout](#).
 - c. **Other Options** to edit form precision and to change which context menus are associated with the form. See [Setting Form Precision and Other Options](#).
 - d. **Business Rules** to change which business rules are associated with the form, or modify business rule properties. See [Using Business Rules](#).
3. Click **Finish** to save your work and close the form.

Editing Composite Forms

To edit composite forms:

1. Take one of these actions:
 - Select the form, click the **Show Usage** icon , and then click **Edit**.
 - Select the form, and then click **Edit** (see [Selecting and Opening Forms and Folders](#)).

Note:

When editing a composite form, if this message displays, "Modifications have been made to one or more included forms; if you want to save changes to common dimensions, save the composite form," determine what changes were made to the common dimensions of the included simple forms before saving changes to the composite form.

2. Select:
 - **Properties** to edit the composite form name, description or instructions. See [Creating Composite Forms](#).
 - **Layout** to edit form layout and properties. See [Setting Composite Form Layout](#).
 - **Business Rules** to change which business rules are associated with the form, or modify business rule properties. See [Using Business Rules](#).
3. Click **Finish** to save your work and close the form.

Moving, Deleting, and Renaming Forms

To move, delete, and rename forms:

1. Select the form.
See [Selecting and Opening Forms and Folders](#).
2. Select a task:
 - To move a form, click **Move**, and then select the destination folder.

 **Note:**

You can move multiple forms simultaneously if they are in the same folder.

- To delete a form, click **Delete**.
 - To rename a form, click **Rename**, and then enter the new name.
3. Click **OK**.

Importing and Exporting Form Definitions

Administrators can use `FormDefUtil.cmd` (Windows) or `FormDefUtil.sh` (UNIX) to move form definitions between Oracle Hyperion Planning applications. You can export or import form definitions to or from an XML file, useful for moving from a development to a production environment.

The utility uses a command line interface and is installed in the `planning1` directory. For the full path to `planning1`, see [About EPM Oracle Instance](#).

 **Note:**

XML files for composite forms that were exported in an earlier release cannot be imported into the current release. You must re-extract the XML files after the corresponding Planning applications have been migrated to the current release. XML files for non-composite forms exported in an earlier release can be imported into the current release.

To launch the `FormDefUtil` utility:

1. Enter the command from the `planning1` directory, using this syntax:

```
formdefutil [-f:passwordFile] import|export filename|formname|-all server name user name application
```

Parameter	Purpose	Required?
<code>[-f:passwordFile]</code>	If an encrypted password file is set up, you can use this option as the first parameter in the command line to run the utility with the full file path and name specified in <code>passwordFile</code> . See Suppressing Password Prompts in Planning Utilities .	No
<code>import export</code>	Import or export the form definition.	Yes
<code>filename formname -all</code>	When used with <code>import</code> , specify the XML file containing the form definition. When used with <code>export</code> , specify the form to export to XML. Use <code>-all</code> with <code>import</code> or <code>export</code> to import or export all XML files or form definitions in the current application.	Yes. <code>-all</code> is optional.

Parameter	Purpose	Required?
server name	Server name on which the Planning application resides.	Yes
user name	Administrator's name.	Yes
application	When used with <code>export</code> , the name of the Planning application containing the form definitions to export. When used with <code>import</code> , the name of the Planning application to which to import the form definition.	Yes

- If prompted, enter your password.

When you export form definitions, the utility creates an XML file in the current directory and writes errors in a log file. (For information on the location of log files, see [About EPM Oracle Instance](#).) You can copy the utility to any directory and launch it from there to save files to another directory.

Table 6-13 FormDefUtil.cmd Command Examples

Task	Example
Importing one file	<code>FormDefUtil.cmd import c:\EPM_ORACLE_INSTANCE\Planning\planning1\form1.xml localhost admin APP1</code>
Exporting one file	<code>FormDefUtil.cmd export Form1 localhost admin APP1</code>
Exporting all form definitions	<code>FormDefUtil.cmd export -all localhost admin APP1</code>
Import all form definitions	<code>FormDefUtil.cmd import -all localhost admin APP1</code>



Note:

Replace `EPM_ORACLE_INSTANCE` with the absolute or relative path to the form file on your system. For more information, see [About EPM Oracle Instance](#).

Specifying How Many Forms Users Can Simultaneously Use

To enable users to work in more than one form at one time, you can update the `DATA_GRID_CACHE_SIZE` property, which specifies the number of data grids cached per user for an application. By default, this property is set to 1, allowing each user to have one form open and active at a time.

▲ Caution:

Keep in mind that increasing this property value causes the application server to use more memory. For example, for an application with 100 users, by default the application server can cache up to 100 data grids. If this property is set to 3, up to 300 data grids can be cached. The impact on memory usage is greater for applications having very large forms.

To specify how many forms each user can simultaneously work in:

1. Select **Administration**, then **Application**, and then **Properties**.
2. To set properties for all Oracle Hyperion Planning applications, select **System Properties**.
3. Update the setting:
 - To add the property, click **Add**. In the blank row, enter `DATA_GRID_CACHE_SIZE` (avoid using spaces). Under **Property Value**, enter a number to represent the number of forms that a user can have open and active at one time.
 - To change the property, change its value under **Property Value**.
 - To delete the property, select its name, and then click **Delete**.
4. Click **Save** and confirm your changes.
5. Restart the Planning server.

Using Grid Diagnostics

Grid diagnostics allow you to view the time it takes to open forms and ad hoc grids. You can select whether to view the load times in chart form or tabular form, and you can select on which forms and grids to run the diagnostics. This helps you to identify forms with poor performance and address design issues in forms.

To use grid diagnostics:

1. Log in to a Oracle Hyperion Planning application.
2. Select **Tools**, then **Diagnostics**, and then **Grids**.
3. Select the forms on which you want to run the diagnostics and click **Run Diagnostics**.

A pie chart is displayed in **Diagnostics Summary** that shows the percentage of load times for forms and ad hoc grids that are greater than five seconds, between one and five seconds, and less than one second.

4. Click on a section of the pie chart to display more detail about a specific load time in the **Display** area under **Diagnostics Summary**

For example, clicking Load Time (Greater than 5s) on the pie chart displays details about the forms with load times greater than five seconds in the Display area.

5. Once you have selected the load time detail to display, select whether to display the information in **Tabular** format or as a **Chart**.

If you select to display the load times as a chart, select the chart type: Area, Horizontal Bar, Bar, or Line.

Using Business Rules

About Business Rules

With appropriate access, users can launch business rules from Oracle Hyperion Planning. Business rules can also prompt users for input when rules are launched.

For optimum performance, business rules that run in forms should be designed to execute within three minutes. For business rules with longer execution time, you can schedule batch processing or run the business rules during non-peak hours.

For information on:

- Creating and updating business rules, see *Oracle Hyperion Calculation Manager Designer's Guide*.
- Using runtime prompts, see [About Runtime Prompts](#).

Selecting Business Rules

On the **Business Rules** tab, you can associate multiple business rules with a form, by plan type. Users can launch associated business rules from the form to calculate and allocate values. You can set whether each business rule associated with a form automatically launches when the form is opened or saved.

To select business rules for forms:

1. Take an action:
 - To update the current form, click **Business Rules**.
 - To open a form for editing, click **Edit**, and then click **Business Rules**.
2. From the **Plan Type** drop-down menu, select the plan type.
3. From the **Business Rules** list, select the business rules to associate with the form, and move them to **Selected Business Rules**. See [Using the Member Selector](#).

By default, the Calculate Form and Calculate Currencies business rules are selected. Calculate Form is automatically created for forms to calculate subtotals. Calculate Currencies is created for forms that include multiple currencies in a row, column, or page, to enable converting values among the available currencies. You can clear Calculate Currencies if you use customized calculation scripts to calculate currency conversions. You can clear Calculate Form to prevent planners from calculating data in forms.

4. To change the order of selected business rules (the order in which rules display and launch), select a business rule in **Selected Business Rules** and click the up or down arrow to move it up or down in the list. The rule listed first displays and launches first; the rule at the bottom of the list displays and launches last.

The order in which business rules launch is important and may affect data. For example, it is important to convert currencies first, before subtotaling.

5. To set business rule properties, click **Properties**. See [Setting Business Rule Properties](#).
6. Click **Save** to save your work and continue creating or editing the form, or click **Finish** to save your work and close the form.

 **Note:**

When selecting business rules for composite forms, you can select which included forms' business rules run in composite forms. Business rules from included forms do not run unless you select them for the composite form itself. For example, to run all the business rules associated with an included form named "Total Expense Impact", select "Business rules for Total Expense Impact".

Setting Business Rule Properties

You can specify whether business rules associated with forms launch automatically when users open or save the form. If business rules have runtime prompts, you can set whether the default members in the runtime prompt match the members selected in the page and Point of View axes.

To set business rule properties:

1. Take an action:
 - To update the current form, click **Business Rules**.
 - To open a form for editing, click **Edit**, and then click **Business Rules**.
2. Click the **Business Rules** tab.
3. Select **Run on Load** next to a business rule to launch it automatically when the form is opened.

Business rules having runtime prompts cannot launch on load.
4. Select **Run on Save** next to a business rule to launch it automatically when saving a form.

If business set to **Run on Save** contain runtime prompts, users are prompted to enter the runtime prompt value before the form is saved.
5. **Optional:** If a business rule has runtime prompts, select **Use Members on Form** to match the default member selection on the runtime prompt window to the current members in the page and Point of View axes of the open form.

To learn how this option interacts with other settings and conditions, see [Understanding Runtime Prompts](#).
6. **Optional:** To hide the runtime prompt value from the user, select **Hide Prompt**, which automatically selects **Use Members on Form**.

After saving the form, the next time you return to this page, **Use Members on Form** displays as selected.

You can hide runtime prompts if:

 - All runtime prompt member values are filled in (appropriate dimension members can be read from form's Page/Point of View)
 - No dimensions are repeated in the runtime prompt
7. Click **OK**.

About Runtime Prompts

When launched, business rules can prompt users for such variables as members, text, dates, or numbers. Prompts should be specific and tell users what type of data is expected. For example:

- Select a month.
- Enter the expected number of customer visits per quarter.
- What percentage change in earnings do you expect next month?

If, in Oracle Hyperion Calculation Manager, the option **Create dynamic members** is selected for the business rule and the parent member is enabled for adding dynamic children as described in [About Dynamic Members](#), then users can create new members by entering their name in the runtime prompt.

When launching business rules with runtime prompts, Oracle Hyperion Planning validates the value entered, but not the business rule. To set the default member selection in a runtime prompt, see [Setting Business Rule Properties](#). To understand how other settings and conditions affect runtime prompts, see [Understanding Runtime Prompts](#).

By default, the values for processed runtime prompts in the application are stored in the database and available for viewing from the Job Console (select **Tools**, then **Job Console**). If many users are running business rules with runtime prompts, tracking these values consumes significant system resources. To improve performance, you can turn off this function so Planning does not capture runtime prompt values. To do so, add the `CAPTURE_RTP_ON_JOB_CONSOLE` property to the properties table, with the property value of `FALSE` (to turn it back on again, change its value to `TRUE`). See [Setting Application and System Properties](#).

Understanding Runtime Prompts

The display and values of runtime prompts are affected by such aspects as:

- Whether the **Use as Override Value** property is set at the rule or the ruleset level at design-time
- Whether there are valid members on the form's Page/Point of View and whether the **Use Members on Form** and **Hide Prompt** options on the **Business Rule Properties** tab are selected (see [Setting Business Rule Properties](#))
- Whether the **Runtime Prompt** option is set during form design or set when designing the runtime prompt (see *Oracle Hyperion Calculation Manager Designer's Guide*)
- Whether the **Use Last Value** property is set when designing the business rule
- Whether, in Oracle Hyperion Calculation Manager, the option **Create dynamic members** is selected for the business rule and the parent member is enabled for adding dynamic children as described in [About Dynamic Members](#). If yes, then users can create new members by entering their name in the runtime prompt.

Principles:

1. If the **Use as Override Value** property is set at the rule or the ruleset level at design-time, the value overridden at the rule level or the ruleset level will take precedence over the values of members in the Page/Point of View and the last saved value. This occurs regardless of where the rule is launched (from the form or the **Tools**, then **Business Rules** menu), and regardless of whether the runtime prompt is hidden during design. The

Override Value can be set as a Oracle Hyperion Planning user variable, in which case the rule will be launched with the current value of the variable.

2. When launched from a form, the values of members in the Page/Point of View take precedence over the last saved value if the **Use Members on Form** option is selected, regardless of whether the runtime prompt is hidden during design. The business rule is run without displaying the hidden runtime prompt to users, and the runtime prompt values are taken from the Page/Point of View members.

This is not the case when business rules associated with composite forms are launched on save or from the left-hand pane or when business rules are launched from the **Tools**, then **Business Rules** menu. In these cases, the **Use Members on Form** setting is ignored, hidden runtime prompts get design-time values, and the last saved value takes precedence.

3. If the **Use Last Value** option is selected for the runtime prompt at design time, and if any of these conditions exist:
 - **Use Members on Form** is not selected
 - A runtime prompt is launched from the **Tools**, then **Business Rules** menu
 - Values cannot be pre-filled from the context

Then the precedence of runtime prompt values is determined by:

- a. The last saved value takes precedence.
- b. If a ruleset is launched, the value overridden at the ruleset level at design-time is used.
- c. If a business rule is launched, the value overridden at the rule-level at design-time is used. If it is not overridden at the rule-level, the runtime prompt value at design-time is used.

Runtime prompts that are hidden at design time never use the last saved value. In these cases, the **Use Last Value** setting is ignored.

4. The **Use Members on Form** and **Hide Prompt** options apply only to Member and Cross Dimension runtime prompt types (Cross Dimension runtime prompt types are available only for business rules created with Calculation Manager).

The value set at the rule or the ruleset level at design-time when the **Use as Override Value** property is set participates in the **Hide Prompt** behavior.

5. For Cross Dimension runtime prompts: the runtime prompt is not hidden unless all the prompts in the runtime prompt can be pre-filled from the **Override Value** or Page/Point of View. The runtime prompt is displayed with some values pre-filled from the **Override Value** or Page/Point of View and others filled according to Principles 1, 2 and 3.

This table describes the result on runtime prompts of these settings and conditions:

Table 6-14 How Member Availability and Other Settings Affect Runtime Prompts

Availability of Override Value and member on the Page/Point of View	Use Members on Form option is selected	Hide Runtime Prompt property is set during runtime prompt design	Hide Prompt option is selected for the form	Result on Runtime Prompt
Use as Override Value is set and Override Value is available or the member is available on the Page/Point of View to use as the runtime prompt value.	Yes	Yes	Yes or No Setting is ignored	The business rule runs without displaying the runtime prompt to users. Instead, the runtime prompt value is taken from the Override Value or Page/Point of View member.
Use as Override Value is set and Override Value is available or the member is available on the Page/Point of View to use as the runtime prompt value.	Yes	No	Yes	If all runtime prompts can be pre-filled from the Override Value or Page/Point of View context and are valid and within limits, the runtime prompts are not displayed. However, if even one runtime prompt value cannot be pre-filled from the Override Value or Page/Point of View context, then all runtime prompts display, with values pre-filled wherever possible. All others follow Principles 1 and 3.
Use as Override Value is set and Override Value is available or the member is available on the Page/Point of View to use as the runtime prompt value.	Yes	No	No	The runtime prompt is displayed to users, with values pre-filled from the Override Value or Page/Point of View.
Use as Override Value is not set and the member is not available on the Page/Point of View to use as the runtime prompt value.	Yes	Yes	Yes or No Setting is ignored	The business rule displays the runtime prompt to users, with values pre-filled according to Principle 3. For example, the form context cannot be passed because the dimension of the runtime prompt is on rows or columns, so the Hide Prompt setting is ignored and the runtime prompt displayed.

Table 6-14 (Cont.) How Member Availability and Other Settings Affect Runtime Prompts

Availability of Override Value and member on the Page/Point of View	Use Members on Form option is selected	Hide Runtime Prompt property is set during runtime prompt design	Hide Prompt option is selected for the form	Result on Runtime Prompt
Use as Override Value is not set and the member is not available on the Page/Point of View to use as the runtime prompt value.	Yes	No	Yes	The runtime prompt is displayed to users, with values pre-filled according to Principle 3.
Use as Override Value is set and Override Value is available, and the member is not available on the Page/Point of View to use as the runtime prompt value.	Yes	No	No	If all runtime prompts can be pre-filled from the Override Value and are valid and within limits, the runtime prompts are not displayed. However, if even one runtime prompt value cannot be pre-filled from the Override Value , then all runtime prompts display, with values pre-filled wherever possible. All others follow Principles 1 and 3.
Use as Override Value is not set and the member is not available on the Page/Point of View to use as the runtime prompt value.	Yes	No	No	The runtime prompt is displayed to users, with values pre-filled according to Principle 3.
Use as Override Value is set and Override Value is available, and the member is not available on the Page/Point of View to use as the runtime prompt value.	Yes	No	No	The runtime prompt is displayed to users, with values pre-filled according to Principles 1 and 3.
Use as Override Value is set and Override Value is available or the member is available on the Page/Point of View to use as the runtime prompt value.	No	Yes	Not available	The business rule runs without displaying the runtime prompt to users. Instead, the design-time values are used.
Use as Override Value is set and Override Value is available or the member is available on the Page/Point of View to use as the runtime prompt value.	No	No	Not available	The runtime prompt is displayed to users, with values pre-filled according to Principle 3.

Table 6-14 (Cont.) How Member Availability and Other Settings Affect Runtime Prompts

Availability of Override Value and member on the Page/Point of View	Use Members on Form option is selected	Hide Runtime Prompt property is set during runtime prompt design	Hide Prompt option is selected for the form	Result on Runtime Prompt
Use as Override Value is not set and the member is not available on the Page/Point of View to use as the runtime prompt value.	No	Yes	Not available	The business rule runs without displaying the runtime prompt to users. Instead, the design-time values are used.
Use as Override Value is not set and the member is not available on the Page/Point of View to use as the runtime prompt value.	No	No	Not available	The runtime prompt is displayed to users, with values pre-filled according to Principle 3.

When hidden runtime prompt values are ambiguous, note:

- If the form context cannot be passed in (because the dimension is on the row or column, for example), hidden runtime prompts are displayed.
- With hidden Cross Dimension runtime prompt types, if all prompts cannot be passed in from the Override Value or context, the runtime prompt displays with values pre-filled from **Override Value** or context values and design time values. For example, if the Cross Dimension has runtime prompts for Period, Entity, and Scenario, and Entity is defined on the row and Scenario has an **Override Value** set, then the runtime prompt displays with the Override Scenario, then design time value for Entity, then Page Period.
- If the **Override Value** is present, or context can be passed in for the runtime prompt value but it is out of limits, then the runtime prompt is displayed with the context value pre-filled.
- If there is more than one runtime prompt of type Member or type Cross Dimension combined, then the runtime prompts are displayed with the **Override Value** or context values pre-filled. For example, if there is a member type runtime prompt for the Entity dimension and a Cross Dimension type runtime prompt with one prompt for the Entity dimension, then both runtime prompts are displayed. This rule does not apply to Calculation Manager rulesets.
- When launched from the **Tools**, then **Business Rules** menu, runtime prompts are hidden and the design-time value (overridden at the rule or ruleset level) is used to launch the business rule. If the provided design-time value is out of limits, then the runtime prompt is displayed with the design-time value pre-filled.
- Runtime variables that are hidden during design never use the last saved value. The **Use Last Value** property is ignored, and the values are not saved to the database.

About Runtime Prompts and Approvals Security

Administrators can design runtime prompts to honor Approvals security for members. Doing so prevents planners from changing data in planning units to which they do not have access, according to Approvals rules. For example, the administrator may not want planners to change data after they have promoted the related planning unit. In Oracle Hyperion Calculation Manager, administrators can set runtime **Security** for a member or members:

- **Approvals:** Oracle Hyperion Planning allows a user to change member data if both these conditions are true:
 - The user has write access to the members (as assigned in Planning).
 - If the members belong to a planning unit, the user owns the planning unit.If both conditions are not met, the user cannot change the members' data.
 - **Write:** Users launching the business rule who have write access to the members (as assigned in Planning) can change its data. The members' Approvals status is ignored.
 - **Read:** Users launching the business rule have read access to the members (as assigned in Planning). Approvals status is ignored.
 - **Use Default:** Security is applied to the runtime prompt only if member access is not set to None (that is, either read or write).
- When business rules are migrated from an earlier release, they assume this setting.

See [Designing Secure Runtime Prompts](#).

Designing Secure Runtime Prompts

Oracle Hyperion Planning supports runtime security on Scenario, Version, Entity, and secondary dimension intersections by relying on the order in which runtime prompts are designed. For the business rule to apply **Approvals** security, the Oracle Hyperion Calculation Manager designer must place runtime prompts for Scenario and Version with Write or Approvals security before the Entity runtime prompt.

Runtime prompts for Version and Scenario members are filtered by write access when **Security** in Calculation Manager is set to either **Approvals** or **Write**. Runtime prompts for Entities with the **Security** set to **Approvals** are filtered according to the last Scenario/Version that is displayed before the Entity runtime prompt. If either the Scenario or Version runtime prompt does not exist, Entities are filtered by write access.

Runtime prompts for other dimensions with **Security** set to **Approvals** are considered to be secondary dimensions and are filtered according to the last Scenario/Version/Entity that displays before that runtime prompt.

So, for runtime prompts using **Approvals** security defined in this order:

Scenario1: Version2: To_Entity1, Scenario2: Version1, To_Entity2, To_Product

Runtime prompts are filtered as follows:

- To_Entity1 is filtered by the combination: Scenario1: Version2
- To_Entity2 is filtered by the combination: Scenario2: Version1
- To_Product is filtered by the combination: Scenario2, Version1, To_Entity2, To_Product

For example:

```
Fix (FY11, Jan, {EntitySalesByCountry}, {MyScenario2}, {MyVersion2})
```

```
Fix ({MyProduct}, {MyCountry})
```

```
{ToAccount} = {FromAccount} * 2;
```

```
ENDFIX
```

```
Endfix
```



Note:

If the business rule designer omits a planning unit dimension from the list of runtime prompts (for example, the runtime prompt does not include a Scenario or Entity), then Approvals security is not applied and the hierarchy is filtered by write access.

See [About Runtime Prompts and Approvals Security](#).

Launching Business Rules With a Utility

Using the `CalcMgrCmdLineLauncher.cmd` utility, administrators can launch—from the Command Prompt—business rules created with Oracle Hyperion Calculation Manager.

Notes:

- Use the `/Validate` option to check the command syntax before you launch the business rule.
- If the launched business rule has a runtime prompt, before running `CalcMgrCmdLineLauncher.cmd`, create a file containing the runtime prompt values. You can either manually create an ASCII file that contains the runtime prompt values, or you can automatically generate the values file by selecting the Create Runtime Prompt Values File option on the Runtime Prompts page.

To launch a business rule with `CalcMgrCmdLineLauncher.cmd`:

1. When launching a business rule having runtime prompts, generate a file containing the runtime prompt values by either:
 - Specifying the name of the runtime prompt values file that you generated on the Runtime Prompts page (see the *Oracle Hyperion Planning User's Guide*).
 - Creating a runtime prompt ASCII file that contains, on separate lines, each runtime prompt name and its value, separated by a double colon (`::`). For example:

```
CopyDataFrom::Jan
```

```
CopyDataTo::Apr
```

This file specifies that the value for the runtime prompt named `CopyDataFrom` is Jan, and that the value for the runtime prompt named `CopyDataTo` is Apr.

Save the file in the `planning1` directory (for the full path to `planning1`, see [About EPM Oracle Instance](#)). Alternately, when executing the utility, specify the full path to the runtime prompt file.

- From the `planning1` directory, enter this command at the Command Prompt, one space, and the parameters, each separated by a space:

```
CalcMgrCmdLineLauncher.cmd [-f:passwordFile] /A:appname /U:username /
D:database [/R:business rule name] /F:runtime prompts file [/validate]
```

Table 6-15 CalcMgrCmdLineLauncher Parameters

Parameter	Purpose	Required?
<code>[-f:passwordFile]</code>	If an encrypted password file is set up, you can use this option as the first parameter in the command line to run the utility with the full file path and name specified in <code>passwordFile</code> . See Suppressing Password Prompts in Planning Utilities .	No
<code>/A:appname</code>	Specify the Oracle Hyperion Planning application from which the business rule is launched	Yes
<code>/U:username</code>	Specify the administrator's user name	Yes
<code>/D:database</code>	Specify the name of the plan type against which the calculation is launched	Yes
<code>[/R:business rule name]</code>	Specify the name of the business rule to be launched	Yes
<code>/F:runtime prompts file</code>	Specify the name of the file containing business rule's runtime prompt names and values	Yes, when launching a business rule having a runtime prompt.
<code>[/validate]</code>	Checks the command syntax only; does not launch the business rule. Any errors are displayed on the console and written to the Calculation Manager log file. If the <code>CalcMgrLog4j.properties</code> file is in the Classpath, the log file is generated in the <code>EPM_ORACLE_INSTANCE/diagnostics/logs/planning</code> directory.	No
<code>/?</code>	Print the syntax and options for <code>CalcMgrCmdLineLauncher.cmd</code>	No

For example to launch the rule named Depreciate, using runtime prompt values in a file called `Values.txt`, enter:

```
CalcMgrCmdLineLauncher.cmd /A:planapp /U:admin /D:plan1 /R:Depreciate /
F:Values.xml
```

- If prompted, enter your password.

Customizing Error Messages

Administrators can use the Oracle Essbase `@RETURN` function to customize displayed messages when a business rule calculation is terminated. For example, customize a business rule to display: "You must specify a maximum value before this calculation can successfully execute."

The business rule syntax:

```
@RETURN ("ErrorMessage",ERROR)
```

where:

- "ErrorMessage" is an error message string, or any expression that returns a string.
- ERROR indicates that the message indicated in the "ErrorMessage" string is displayed to the user, the Job Console, and the application log as an error type message.

Notes:

- The business rule stops executing when `@RETURN` is called.
- You can use the `IF...ELSEIF` calculation command block to specify logical error conditions, and use the `@RETURN` function to exit the calculation with customized error messages and levels.
- You can also use the following syntax to display error message labels defined in the `HspCustomMsgs` localized resource file:

- Without parameters:

```
@RETURN (@HspMessage ("MESSAGE_LABEL_NAME"), ERROR)
```

- With parameters:

```
@RETURN (@HspMessage (@NAME ("MESSAGE_LABEL_NAME", "PARAM_NAME1",  
"PARAM_VALUE1", "PARAM_NAME2", "PARAM_VALUE") 0, ERROR)
```

For instructions on customizing the `HspCustomMsgs` file, see [Customizing Text, Color, and Images](#).

- You cannot use the function in member formulas.

For more information on the `@RETURN` function, see the *Oracle Essbase Technical Reference*.

How Cell Formatting in Smart View Persists in Planning

With the caveats described in this section, the cell formatting that you apply in Oracle Smart View for Office is applied to the same cells in Oracle Hyperion Planning forms.

Administrators can set an application property, `SMART_VIEW_MERGE_FORMATTING`, that controls whether cell formatting that administrators set in Smart View merge with the cell formatting that non-administrators set in Smart View. See [Controlling Smart View Cell Formatting in Planning Forms](#).

Formatting rules depend on whether the cell formatter is an administrator or a non-administrator:

- **Administrators:** All application administrators share the cell formatting that is saved by other administrators. Their formatting is additive. When an administrator applies different

formatting to the same cells, the latest changes are displayed in the Planning form. If an administrator clears the cell formatting in Smart View, all formatting settings that non-administrators have saved are also cleared.

- **Non-administrators:** When users format cells and save the formatting in Smart View, they see their formatting applied in Planning forms. If users do not save formatting in Smart View, they see the administrators' formatting in Planning forms. Note that because of security and other filtering options, the displayed formatting may not exactly match the administrators' saved formatting. Users can either modify the administrators' formatting and save it as their own formatting settings, or clear their own formatting settings, reverting to the administrators' formatting settings.

To select the formatting for a form:

1. In the form, right-click, and then select **Apply**.
2. Select:
 - **Cell Styles:** To use Planning's formatting
 - **Custom Styles:** To use the formatting saved in Smart View

For more information:

- On which Microsoft Excel formatting features are supported in Smart View and Planning, see "About Smart View Formatting in Planning Forms" in the *Oracle Hyperion Planning User's Guide*.
- On saving Excel formatting, see the *Oracle Smart View for Office User's Guide*.

Working with Substitution Variables

About Substitution Variables

Substitution variables act as global placeholders for information that changes regularly. For example, you could set the current month member to the substitution variable `CurMnth` so that when the month changes, you need not update the month value manually in the form or the report script. You create and assign values to substitution variables within Oracle Hyperion Planning. These substitution variables are then available in Planning when you select members for a form.

You can also create and assign values to substitution variables using Oracle Essbase Administration Services Console or ESSCMD.

For more information about selecting substitution variables, see [About Selecting Substitution Variables as Members](#).

Creating and Assigning Values to Substitution Variables Using Planning

To create and assign values to substitution variables using Oracle Hyperion Planning:

1. Select **Administration**, then **Manage**, then **Variables**.
2. Select the **Substitution Variables** tab.
3. Click **Actions**, then select **Add**.
4. On the **Add Substitution Variable** page, select the **Plan Type**.
5. For **Name**, enter the name of the substitution variable.

6. For **Value**, enter a value for the substitution variable.
7. Click **OK**.

Deleting Substitution Variables Using Planning

To delete substitution variables using Oracle Hyperion Planning:

1. Select **Administration**, then **Manage**, then **Variables**.
2. Select the **Substitution Variables** tab.
3. Select the substitution variable to delete.
4. Click **Actions**, then select **Delete**.
5. Click **Yes**.

Working with User Variables

About User Variables

User variables act as filters in forms, enabling planners to focus only on certain members, such as a department. Before you can associate a user variable with a form, you must create the user variable. When you create forms with user variables, planners must first select values in preferences for the variable before opening forms. Afterward, planners can change the variable on the form only if it is a dynamic user variables. Otherwise, they must continue to set the variable in preferences. For example, if you create a user variable called Division, planners must select a division before working in the form.

The first time planners select a variable for a form, they do so in preferences. Afterward, they can update the variable in preferences or in the form. For information about selecting user variables as members, see [About Selecting User Variables as Members](#).

Managing User Variables

You can set user variables to limit the number of members displayed on a form, helping users focus on certain members. For example, if you create a user variable called Division for the Entity dimension, users can select a member for their own division. You can create any number of user variables for each dimension, and select user variables for any axis in the form. See [Defining Simple Form Page and Point of View](#).

The typical sequence of steps:

1. If necessary, create the appropriate parent-level members in the dimension outline.
2. Define user variables for each dimension you want users to be able to filter.
See [Creating User Variables](#).
3. When designing the form, associate the user variable with the form.
See [About Selecting User Variables as Members](#).
4. Instruct users to select a member for the user variable associated with the form.

Before users can open forms that have user variables, they must select a member for User Variable Options in preferences. After selecting an initial value, they can change it in the form or in preferences. See *Oracle Hyperion Planning User's Guide*.

Creating User Variables

To create user variables:

1. Select **Administration**, then **Manage**, then **Variables**.
2. Select the **User Variables** tab.
3. Click **Actions**, then select **Add**.
4. In the **User Variables** window, for **Dimension Name**, select the dimension for which to create a user variable.
5. For **User Variable Name**, enter the name of the user variable.
6. **Optional:** Select **Use Context** to allow user variables to be used in the Point of View. With this setting, the value of the user variable changes dynamically based on the context of the form.
7. Click **OK**.

You can now associate the user variable with a form. See [About Selecting User Variables as Members](#). Afterward, planners can select members for the user variable. See *Oracle Hyperion Planning User's Guide*.

Deleting User Variables

To delete user variables:

1. Select **Administration**, then **Manage**, then **Variables**.
2. Select the **User Variables** tab.
3. Select the user variable to delete.
4. Click **Actions**, then select **Delete**.
5. Click **Yes**.

7

Using the Member Selector

About Working with Members






Use the Member Selection dialog box to select members and other information to use with features such as forms and business rule runtime prompts. If variables and attributes are defined, you can also select variables and attributes. You can display and select members by member name, alias, or both. The display options that you define for the Member Selection dialog box override those defined as an application default by an Administrator, and those specified as an application preference.




Selecting Members

Use the Member Selection dialog box to select members and other information for Oracle Hyperion Planning features. For example, you can select members for form management, business rule runtime prompts, and Clear Cell Details. You can select members by member name, alias, or both, depending on the setting for the member in the dimension editor for the current application and member selection options set by users.



To select members:

1. From the members list, select members.
2. **Optional:** Select or enter search criteria.
 - a. In the Search box, select options: **Member Name**, **Member Alias**, **Description**, or **UDA**.

You can also enter all or part of a member name, alias, description, or UDA. Searches include shared members. To search for an exact match, select **Exact** and enter the search string. The search is not case-sensitive. You can search for a word, multiple words, or wildcard characters. See [Using Wildcards in Searches](#).
 - b. Click .
3. **Optional:** To set how information displays or filter members in the left pane of the Member Selection dialog box for the current session, click one of the following icons above the members list. You can select multiple filters, such as functions and generations.
 -  Collapse all members
 -  Expand all members
 -  Display Properties. Select **Member Name**, **Alias**, **Member Name: Alias**, or **Alias: Member Name**. Or, select **Additional Display Properties**, then **Description** or **Count**. Selected options are displayed in columns in the left and right panes. To remove columns, clear the selections.
 -  Keep only by Functions. Select **Member**, **Descendants**, **Descendants (inc)**, **Ancestors**, **Ancestors (inc)**, **Siblings**, **Siblings (inc)**, **Parents**, **Parents (inc)**, **Children**, **Children (inc)**, or **Level 0 Descendants**.

-  Keep only by Attributes. If attributes are defined, select options for **Attributes**, **Operator**, and **Value**, and then click **OK**. Multiple attributes can be selected.
 -  Keep only by levels or generations. Select **Level** or **Generation**, select the levels or generations to display, and then click **OK**.
 -  Refresh the display to remove filters.
4. From the members list, make a selection.
Only members to which you have access are displayed.

Notes:

- **For runtime prompts only:** The displayed hierarchy is for the application and plan type against which the business rule is launched. For runtime prompts, members are displayed that satisfy runtime prompt limits.
 - **For business rules or validation rules having Cross Dimension or Member Range runtime prompts only:** From the **Select Dimension** list, select a dimension that the business rule designer set for this runtime prompt.
 - **For shared members only:** Shared members are displayed in this format: *shared_member.parent_member* (shared). When they are selected on the right side of the dialog box, only the shared member name is displayed.
5. **Optional:** Click  or  to collapse or expand the hierarchy.
6. Move members to or from the **Selected Members** list.



Click  in the middle of the dialog box to select a range of members based on hierarchy relationships.

Table 7-1 Member Relationships

Relationship	Members Included
Member	The selected member
Descendants	All descendants of the selected member, excluding the selected member
Descendants (inc)	The selected member and its descendants
Ancestors	All members above the selected member, excluding the selected member
Ancestors (inc)	The selected member and its ancestors
Siblings	All members from the same level in the hierarchy as the selected member, excluding the selected member
Siblings (inc)	The selected member and its siblings
Parents	The member in the level above the selected member
Parents (inc)	The selected member and its parent
Children	All members in the level immediately below the selected member
Children (inc)	The selected member and its children
Level 0 Descendants	All descendants of the selected member that have no children

Optional: Click the Function Selector icon  on the right side of the dialog box to insert functions for the members selected in the right pane.

7. If variables or attributes are defined, click the **Variables** tab to select members for user variables, substitution variables, and attributes. Members of each category are displayed as children. Only members to which the user has read access are displayed in forms.
 - **User Variables:** Select members as described in [About Selecting User Variables as Members](#).
 - **Substitution Variables:** Select members as described in [About Selecting Substitution Variables as Members](#). The selection depends on the preference setting for **Enable the Display of Substitution Variables**, described in [Specifying System Settings](#).
 - **Attributes:** Select attributes based on the values described in the following table. Selecting a non-level 0 attribute selects all level 0 descendants and applies the operator to each, as described in [About Selecting Attribute Values as Members](#).


Optional: To select functions that apply to custom attributes, click . To filter by dimension, select an option from the Dimension drop-down list.

Table 7-2 Attribute Value Selection

Operator	Attribute Values Included
Equal	Equal to the selected attribute
NotEqual	Not equal to the selected attribute
Greater	Greater than the selected attribute
GreaterOrEqual	Greater than or equal to the selected attribute
Less	Less than the selected attribute
LessOrEqual	Less than or equal to the selected attribute

8. Be sure the member selection options are appropriate for the context from which you invoked the member selector dialog box.
9. Click **OK**.

Using Wildcards in Searches

You can use these wildcard characters to search for members.

Table 7-3 Wildcard Characters

Wildcard	Description
?	Match any single character
*	Match zero or multiple characters. For example, enter "sale*" to find "Sales" and "Sale" because * includes zero or more characters after the word "sale." The default search uses the * wildcard. For example, entering "cash" searches for "*cash*" and returns "Restricted Cash", "Cash Equivalents", "Cash", and "Noncash Expenses" because the word "cash" appears within each matched item.

Table 7-3 (Cont.) Wildcard Characters

Wildcard	Description
#	Match any single number (0-9)
[list]	Match any single character within a specified list of characters. You can list specific characters to use as wildcard. For example, enter [plan] to use all the letters within the brackets as a single wildcard character. You can use the "-" character to specify a range, such as [A-Z] or [!0-9]. To use the "-" character as part of the list, enter it at the beginning of the list. For example, [-@&] uses the characters within the brackets as wildcard characters.
[!list]	Match any single character not found within a specified list of characters. The "-" character can also be used to indicate a range, such as [!A-Z] or [!0-9].

About Selecting Attribute Values as Members

If attribute members are defined, you can select attribute values on the Member Selection page. For attribute members, selecting a non-level 0 attribute selects all level 0 descendants and applies the operator to each. For attributes of type numeric, date, and Boolean (where false = 0 and true = 1), evaluation is based on the minimum and maximum values. For text attributes, evaluation is based on the position from top to bottom in the hierarchy. The top position has the lowest value, and the bottom position has the highest value.

Example: Numeric attribute

In this example, the selected operator is applied to each level 0 descendant, based on the numeric value. For example, selecting NotEqual and Small in the Member Selection page includes all values not equal to 1 and not equal to 2, so the selection includes 3, 4, 5, and 6. Selecting Greater and Small includes all values greater than 1 or greater than 2, so the selection includes 2, 3, 4, 5, and 6.

Size

Small

1

2

Medium

3

4

Large

5

6

Table 7-4 Example: Numeric Attribute Evaluation

Selected Operator	Selected Attribute Value	Result	Explanation
Equal	Large	5, 6	The Equal operator is applied to all level 0 descendants of Large, which includes 5 and 6.
Less	Medium	1, 2, 3	The Less operator is applied to all level 0 descendants of Medium. This includes values < 3 OR < 4, which results in 1, 2, and 3.
Greater	Medium	4, 5, 6	The Greater operator is applied to all level 0 descendants of Medium. This includes values > 3 OR > 4, which results in 4, 5, and 6.
GreaterOrEqual	Medium	3, 4, 5, 6	The GreaterOrEqual operator is applied to all level 0 descendants of Medium. This includes values >=3 OR >= 4, which results in 3, 4, 5, and 6.
LessOrEqual	Medium	1, 2, 3, 4	The LessOrEqual operator is applied to all level 0 descendants of Medium. This includes values <=3 OR <=4, which results in 1, 2, 3, and 4.
NotEqual	Medium	1, 2, 5, 6	The NotEqual operator is applied to all level 0 descendants of Medium. This includes values not equal to 3 AND not equal to 4, which results in 1, 2, 5, and 6.

Example: Text attribute

For text attributes, the selected operator is applied to each level 0 descendant based on its position in the hierarchy, from top (lowest value) to bottom (highest value).

In this example, Envelope is at the top position and has the lowest value. Packet has the next higher value, followed by Box, Carton, Barrel and Crate. Crate is at the bottom position and has the highest value.

For this text attribute, selecting Less and Small includes values that are less than Envelope or less than Packet. Because Envelope is less than Packet, the resulting selection includes only Envelope. Likewise, selecting Greater and Large includes values that are greater than Barrel or greater than Crate, so the resulting selection includes only Crate.

Containers

Small

Envelope

Packet

Medium

Box
Carton
Large
Barrel
Crate

Table 7-5 Example: Text Attribute Evaluation

Selected Operator	Selected Attribute Value	Result	Explanation
Equal	Medium	Box, Carton	The Equal operator is applied to all level 0 descendants of Medium, which includes Box and Carton.
NotEqual	Medium	Envelope, Packet, Barrel, Crate	The NotEqual operator is applied to all level 0 descendants of Medium. This includes values not equal to Box AND not equal to Carton, which results in Envelope, Packet, Barrel, and Crate.
Less	Medium	Box, Packet, Envelope	The Less operator is applied to all level 0 descendants of Medium. This includes everything at a lower position than Carton OR a lower position than Box, which results in Box, Packet, and Envelope.
LessOrEqual	Medium	Envelope, Packet, Box, Carton	The LessOrEqual operator is applied to all level 0 descendants of Medium. This includes everything at the same position as Carton OR at a lower position than Carton, which results in Envelope, Packet, Box, and Carton.

About Selecting Members for Forms

When selecting members for forms:

- To filter members from certain users, restrict their access permissions to members, and then refresh the plan, see [Assigning Access to Members and Business Rules](#).
- The order of members in the **Selected Members** list determines the order on forms. To change the order, select a member and click the Up or Down Arrow above the selected members list.

 **Note:**

If you select members individually and select their parent first, the parent displays in the form at the top of its member hierarchy. (Note that depending on the number of hierarchy levels, calculating totals for the parent of individually selected members could take several passes, slowing calculations). The parent of members selected by relationship, for example, by I(Descendants), displays at the bottom of the hierarchy.

- In the **Layout** tab of the **Form Management** dialog box, you can open the **Member Selection** dialog box by clicking the member selection icon, or by right-clicking a row or column and selecting **Select Members**.
- To select different sets of members across the same dimension, see [Creating Asymmetric Rows and Columns](#).
- For forms with multiple dimensions in a row or column, you can set member selection options for a dimension by selecting that dimension from the **Dimensions** drop-down list that is displayed in the Member Selection dialog box for multiple dimensions in a row or column.
- If you click the member selection icon, an option is displayed for **Place Selection in Separate Rows** or **Place Selection in Separate Columns**. This adds the selection to the rows or columns after the last existing row or column on the form. For example, for a form that contains members Acct1, Acct2, and Acct3 in Column A, if you select these members with **Place Selection in Separate Columns** selected, Acct1 is selected for column A, Acct2 for column B, and Acct3 for column C. If you select the members without this option, all of the members are selected for column A.

This feature is available for single members only, not for members selected with functions, such as Children (inc). For example, if you select Q/IChildren for Column A and select **Place Selection in Separate Columns**, the form layout is not changed.
- All settings except Count are retained after the Member Selection dialog box is closed, and members in the Member Selection dialog box are displayed based on user-defined settings. Members displayed on the Layout tab do not inherit the display settings defined in Member Selection dialog box. Instead, they are displayed using the member name.
- To define different sets of members for a dimension, see [Creating Asymmetric Rows and Columns](#).
- To set display, functionality, and printing options, see [Setting Form Precision and Other Options](#).
- To set up reports that include forms or form definitions, see [Customizing Reports](#).

About Selecting Substitution Variables as Members

Substitution variables act as global placeholders for information that changes regularly. Substitution variables are especially useful for developing and reporting on rolling forecasts. When you select substitution variables as members on the form, their values are based on dynamically generated information. For example, you could set the current month member to the substitution variable `CurMnth` so that when the month changes, you need not update the month value manually in the form or the report script.

Each variable has an assigned value that can be changed centrally on the Oracle Essbase server.

Notes:

- When you open or calculate values on forms, the Oracle Hyperion Planning application replaces substitution variables with values assigned to them.
By default, each substitution variable is retrieved and cached from the Essbase server every five minutes (or 300 seconds). You can change the retrieval interval by adding the `SUBST_VAR_CACHE_LIFETIME` application property and setting its value in seconds.
- You create and assign values to substitution variables within Planning. These substitution variables are then available in Planning when you select members for a form. For instructions on creating and assigning values to substitution variables using Planning, see [Working with Substitution Variables](#).
You can also create and assign values to substitution variables using Oracle Essbase Administration Services Console or ESSCMD.
- Substitution variables must be appropriate for the context in forms. For example, you could select a substitution variable named `CurrQtr` with a value of `Qtr2` as a member of the Time Period dimension. It is not valid to select a substitution variable named `CurrYr` for the Year dimension if its value is `Feb`. You can set substitution variables at the application or database level.
You can also set substitution variables at the Essbase server level.
The same substitution variable can exist on multiple levels; Planning uses the first one it finds as it searches in this order:
 1. Database
 2. Application
 3. Server
- You can select from substitution variables if they are enabled for runtime prompts in business rules, and their values match a member set in the runtime prompt for a business rule.
- Planning checks the validity of substitution variables when they are used (for example, when the form is opened). It does not check when you design forms, so you should test substitution variables by saving and opening forms.
- For information on errors generated when substitution variables are calculated, you can check several logs. See the Essbase server log for information on attempts to use a substitution variable that is no longer contained in Essbase. See the Planning log for information on substitution variables that are not valid in the form. For information about logs, see the *Oracle Enterprise Performance Management System Installation and Configuration Troubleshooting Guide*.

To specify substitution variables in forms:

1. Create the form (see [Creating Simple Forms](#)).
2. In **Member Selection**, select substitution variables the same way you select members, using the right, left, and double arrows to move substitution variables to and from **Selected Members**.

When selected, a substitution variable is preceded by an ampersand (&). For example:

```
&CurrentScenario
```

3. Click **OK**.

About Selecting User Variables as Members

User variables act as filters in forms, enabling planners to focus only on certain members, such as a department. Before you can associate a user variable with a form, you must create the user variable. See [Managing User Variables](#).

When you create forms with user variables, planners must select values for the variable before opening forms. For example, if you create a user variable called Division, planners must select a division before working in the form. The first time you select a variable for a form, you do it in preferences. Afterward, you can update the variable in preferences or in the form.

To select user variables for forms:

1. Create the form (see [Creating Simple Forms](#)).
2. On **Member Selection**, select user variables the same way you select members, using the arrows to move substitution variables to and from **Selected Members**.

User variables are displayed for the current dimension. For example, user variables for the Entity dimension might display as follows:

```
Division = [User Variable]
```

When selected, a user variable is preceded by an ampersand. For example:

```
Idescendants(&Division)
```

3. Click **OK**.

8

Managing Data Validation

Creating and Updating Data Validation Rules

To implement business policies and practices, you can build data validation rules that are checked when conditions are met in forms. Validation messages can be generated if entered data violates validation rules. You can also build limits on submitted planning unit data using validation rules, and designate a reviewer, owner, or notifier to review data that meets some condition.

For example, data validation can ensure that a department's capital expenses adhere to company policies by preventing planners from submitting budgets that contain capital expenditures that fall outside the company's guidelines. Sample scenarios that can be addressed using data validation rules are described in [Data Validation Rule Scenarios](#).

Defining data validation rules involves these main tasks:

- Identifying the data cells or location that you want to display with validation messages or in different colors when conditions are met.
- Identifying the cells that need to participate during rule evaluation, and defining the rule accordingly.
- Creating the data validation rule at the location identified, as described in this topic.

To create and update validation rules:

1. Select **Administration**, then **Manage**, and then **Forms and Ad Hoc Grids**. Create or edit a form, and then click the **Layout** tab in the Form Management page.
2. In the **Layout** tab, right-click the grid, row heading, column heading, or cell for which you want to add or update the validation rule.

Note:



When you hover the cursor over cells in the **Layout** tab, a context menu displays whether the cell contains a validation rule. To view the validation message, select **Show Data Validation Messages**. The context menu also displays when a single cell is selected.

3. Select **Add/Edit Validation Rules** to create or update rules.
4. To add a rule, click **Add Rule** and enter a name and description for the rule.
If necessary, move the rule by selecting an option from **Location**. To create a rule similar to an existing rule, click **Duplicate** and then update the rule. To view rules, click **View Rule**. See [Viewing Data Validation Rules](#).
5. Update the rule.
 - a. Under **Condition**, click and select an option to begin the condition statement: **If**, **Else If**, **Else**, **Then**, **Check Range**, or **Range**.

The first part of a rule must include an If condition. Rules must also contain some form of Then condition. See [Conditions Supported by the Rule Builder](#).

- b. From **Source Type**, select an option for evaluation by the rule.

The **Source Type** list displays the appropriate options for the Condition. For example, If conditions can include **Current Cell Value**, **Cell Value**, **Column Value**, **Row Value**, **Member Name**, **Member**, **Cross-Dim Member**, **Account Type**, **Version Type**, **Var Reporting Type**, **UDA**, or **Attribute**. For detailed information about each type of condition, see [Conditions Supported by the Rule Builder](#).



- c. If applicable for the selected **Source Type**, enter a value in **Source Value** by clicking  to select an option, or  to enter a free form value.

- d. Select the appropriate operator for the evaluation: =, !=, <, <=, >, >=, **Equals**, **Not Equals**, **Contains**, **Starts With**, or **Ends With, In, or Not In**.

For examples, see [Data Validation Conditional Operators](#).








- e. Select an option for the appropriate **Target Type** for the rule.

- f. Update conditions by clicking an icon in the **Actions** area on the right side of the condition builder:

-  Add a condition next to the current row.
-  Delete a condition at the current row.


- g. Select conditions or condition blocks to update.

To update condition blocks, click an icon in the **Condition** area, at the top of the condition builder:

-  Add a condition block within the validation rule starting with If. You can expand or collapse the condition. See [Conditions Supported by the Rule Builder](#).
-  Delete the selected condition block.
-  Delete selected conditions, and copy them to paste in a new location.
-  Copy selected conditions.
-  Paste selected conditions to a new location.
-  Group the selection within a condition, and add a grouping parenthesis. In addition to groupings that you set, the If statement in the grouping block is grouped when a condition is grouped, and the **Custom Grouping** option is enabled.
-  Ungroup the selected conditions. The grouping parentheses are removed from the selected condition. One grouping is removed from the condition each time Ungroup is selected.

You can also set your own grouping for conditions by selecting **Custom Grouping**, then setting up the grouping in the rule definition area.




Selected conditions are displayed as shaded. To clear selected conditions, click once more to the left of the **Condition** column.

6. Click the Process Cell icon  in the rightmost column to add processing instructions. See [Formatting Cells and Setting the Promotional Path](#).

7. When you are ready to enable the rule to make it available in the form, select **Enable Validation Rule**.

 **Tip:**

While you are building a rule, you can save the rule without enabling it. After any errors are resolved and the rule is ready to be validated and used, you can enable and save the rule to make it available in the form. You can temporarily disable a rule by clearing **Enable Validation Rule**.

8. When you finish updating the rule, validate the rule.
 - a. Click **Validate**.
 The validation status displays at the top of the dialog box. You must correct errors before you save changes. If you close the dialog box without validating rules and fixing errors noted during validation, updates are not saved.
 - b. After fixing any errors noted during validation, ensure that **Enable Validation Rule** is selected above the rule definition area to enable the rule for the application.
 - c. After the rule is validated, click **OK**.
9. **Optional:** In the Form Management page, view and update rules.
 - In the **Validation Rules** pane on the right side of the Form Management page, add, edit, or delete rules by clicking , , or .
 - To view rules at the current level or higher, click in the Manage Form page, and then select an option from the Validation Rules drop-down list.
 - If multiple rules are defined at the same location, you can change the order in which rules are processed when rules have the same precedence. To move a rule up, down, or to the top or bottom of the list, select the rule and click the arrows. See [Order of Evaluation and Execution for Data Validation Rules](#).
 - To prevent validations associated with the form from executing when validating the planning unit if the currently logged in user does not have access to the form, select **Validate only for users with access to this form**.
10. In the Form Management page, preview and validate the form, resolve any validation errors, and then save changes.


For forms that have data validation rules enabled, rules are validated when the form is loaded or saved. Data validation rules are saved when the form is saved. See [Creating Simple Forms](#).

When users open the form, they can view and resolve validation messages using the Data Validation Messages pane. See *Oracle Hyperion Planning User's Guide*.

Formatting Cells and Setting the Promotional Path

After a rule is set up, use the Process Cell dialog box to set how cells display in forms, and update the promotional path based on data validations.



To format cells and set the promotional path:

1. In the Data Validation Rule Builder dialog box, click the Process Cell icon  in the rightmost column.

If the icon is not displayed, ensure that the rule is valid and that it permits cell processing instructions. For example, cell processing instructions are included for Else, Range, and Then conditions. The icon is not displayed until all required columns are selected for a rule. See [Creating and Updating Data Validation Rules](#).

2. In the Process Cell dialog box, set how the cell should appear in forms if the conditions defined by this rule are fulfilled.

When the rule is enabled, it does not validate unless you specify at least one of these options: a cell background color, a validation message, or a promotional path option.

- To add or update the cell background color, click . To remove the cell background color, click .
- To display a validation message for the cell, enter the message in the **Validation Message** field. Users see this text when they select **Show Data Validation Messages** in the context menu that appears when you hover over the cell in the form. It also appears as a link in the **Data Validation Messages** pane if data cells are flagged by the validation rules and the **Display message in the Data Validation Messages pane** check box is selected. For information on viewing and resolving data validation errors, see *Oracle Hyperion Planning User's Guide*.
- To update the planning unit promotional path based on the data validation rule specified for the cell, select an Approvals option.

You can leave the option as **None** to specify no changes to the promotional path, if the data cell is in a technically valid state and you are only updating the cell background color or specifying a validation message. You can also select **Update Promotional Path** or **Do Not Promote** (to prevent the planning unit from being promoted if the condition is fulfilled). See [Modifying the Planning Unit Promotional Path](#).

3. Click **OK**.

Updates for the rule are displayed in the Process column for the rule. If you specified a cell color, that color is displayed. You can preview a validation message by hovering the cursor over the Process column.

Viewing Data Validation Rules

After data validation rules are set up with processing instructions, you can use the View Rule dialog box to view all rules that apply to the selected grid, row, column, or cell.

To view data validation rules:

1. In the Data Validation Rule Builder dialog box, click **View Rule** to view all the rules at this level (grid, row, column, cell) for this location in the form.
2. Select the rule name, and then double-click the rule or click **OK** to view details.

For additional information about data validation rules, see:

- [Creating and Updating Data Validation Rules](#)
- [Formatting Cells and Setting the Promotional Path](#)
- [Order of Evaluation and Execution for Data Validation Rules](#)
- [Conditions Supported by the Rule Builder](#)
- [Data Validation Conditional Operators](#)
- [Data Validation Rule Scenarios](#)

Order of Evaluation and Execution for Data Validation Rules

For data validation rules in forms, the precedence for rule evaluation depends on condition priority, location of the rule, and position of the rule in the rule list (if multiple rules exist in the same location). First, cell-level rules are processed. Next, rules at the column level are processed, and then row-level rules are processed. Finally, rules at the grid level are processed. The rules are evaluated based on their position in the rule list within each level.

Location and position determine the order in which the rule will be processed. However, the priority of the processing instructions determines which rule is applied to the data cell. So, if a cell-level rule contains processing instructions with priority 4 and a grid-level rule contains processing instructions with priority 5, the grid-level rule is applied to the data cell. If all rules have processing instructions with the same priority, the first processed rule wins. The priority is based on whether the cell processing instructions specify a validation message, a color, the Do Not Promote promotional path option, or a combination of these settings.

Table 8-1 Priority for Rules in Forms

Default Condition Priority	Validation Message	Color	Do Not Promote
1 (lowest)	X		
1		X	
1	X	X	
2			X
3	X		X
4		X	X
5 (highest)	X	X	X

For additional information about data validation rules, see:

- [Creating and Updating Data Validation Rules](#)
- [Formatting Cells and Setting the Promotional Path](#)
- [Viewing Data Validation Rules](#)
- [Conditions Supported by the Rule Builder](#)
- [Data Validation Conditional Operators](#)
- [Data Validation Rule Scenarios](#)

Conditions Supported by the Rule Builder

These conditions are supported by the data validation rule builder: If, Else, Else If, Then, Check Range, and Range.

For details and examples of the values supported by these conditions, see these sections:

- **If, Else, Else If:** [If Condition Values](#).
- **Then:** [Then Condition Values](#).
- **Check Range, Range:** [Range Condition Values](#).

For additional information about data validation, see:

- [Creating and Updating Data Validation Rules](#)
- [Order of Evaluation and Execution for Data Validation Rules](#)
- [Formatting Cells and Setting the Promotional Path](#)
- [Viewing Data Validation Rules](#)
- [Conditions Supported by the Rule Builder](#)
- [Data Validation Conditional Operators](#)
- [Data Validation Rule Scenarios](#)

If Condition Values

These values are supported by the data validation rule builder for If conditions:

- [Current Cell Value](#)
- [Cell Value](#)
- [Column Value](#)
- [Row Value](#)
- [Cross Dim Member](#)
- [Member Name](#)
- [Member](#)
- [Account Type](#)
- [Version Type](#)
- [Variance Reporting Type](#)
- [UDA](#)
- [Attribute](#)

For information on other conditions, see [Conditions Supported by the Rule Builder](#).

For additional information about data validation, see:

- [Creating and Updating Data Validation Rules](#)
- [Order of Evaluation and Execution for Data Validation Rules](#)
- [Formatting Cells and Setting the Promotional Path](#)
- [Viewing Data Validation Rules](#)
- [Conditions Supported by the Rule Builder](#)
- [Data Validation Conditional Operators](#)
- [Data Validation Rule Scenarios](#)

Current Cell Value

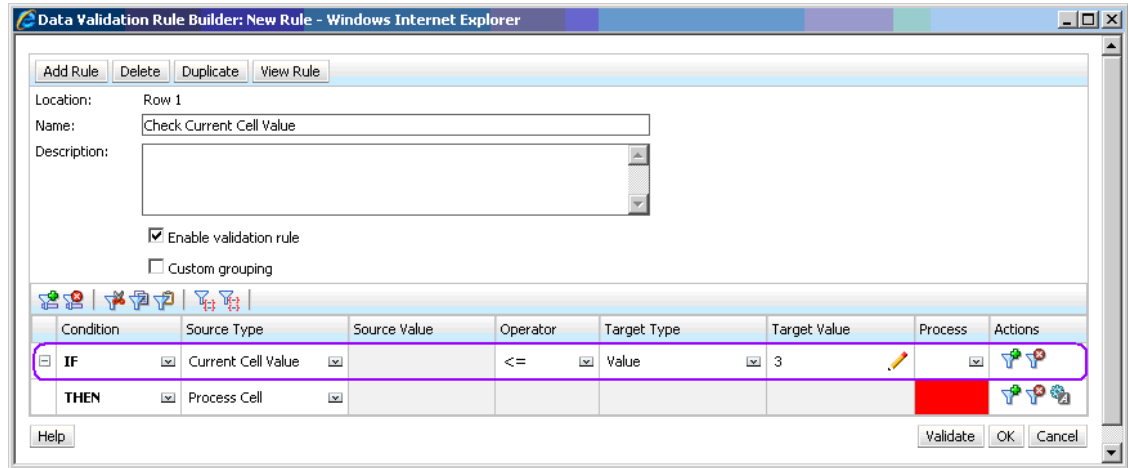
Action:

The action is performed when the value in the current data cell on which the rule is invoked satisfies this condition.

Operators:

The operators available for this function: =, !=, <, <=, >, >=, Equals, Not Equals, Contains, Starts With, or Ends With. These operators act on the selected target value, which can be a free form value, cell value, column value, row value, or cross-dimension member.

Condition Definition:



Condition Evaluation:

With the condition shown in the previous figure, the cells in Row 1 with member Row_Member1 will turn red when the condition is evaluated.

		A	
		Column_Member 1	Column_Member 2
1	Row_Member 1	1.0	2.0
	Row_Member 2	5.0	6.0
	Row_Member 3	9.0	10.0
2	Row_Member 4	13.0	14.0
	Row_Member 5	17.0	18.0

Cell Value

Action:

The action is performed when the value for the specified cell satisfies the condition.

Operators:

The operators available for this function: =, !=, <, <=, >, >=, Equals, Not Equals, Contains, Starts With, Ends With. These operators act on the target value selected, which can be a free form value, cell value, column value, row value, or cross-dimension member.

Condition Definition:

IF Cell Value A,1 = Value 33

Condition Evaluation:

A design-time cell can expand to one or more data cells at data entry time, as shown in the following figure. The value for the cell is the sum of values in all the expanded data cells. For example, the value for cell A1 is the sum of the values in the cells outlined in purple (1+2+5+6+9+10=33), and the value for cell A2 is the sum of the values in the cells outlined in blue (13+14+17+18=62).

		A	
		Column_Member 1	Column_Member 2
1	Row_Member 1	1.0	2.0
	Row_Member 2	5.0	6.0
	Row_Member 3	9.0	10.0
2	Row_Member 4	13.0	14.0
	Row_Member 5	17.0	18.0

Column Value

Action:

The action is performed when the value for the specified column satisfies the condition.

Operators:

The operators available for this function: =, !=, <, <=, >, >=, Equals, Not Equals, Contains, Starts With, or Ends With. These operators act on the target value selected, which can be a free form value, cell value, column value, row value or cross dim member.

Condition Definition:

IF Column Value A < Value 3

Condition Evaluation:

A design-time column can expand to data cells at data entry, as shown in the following figure. The value for a column is the sum of the values in all the expanded data cells in that column at the current row location. The current row changes as the current cell for which the rule is being evaluated changes within the grid.

For example, the value for column A is the sum of the values in the cells outlined in purple (1+2=3), when the rule is evaluated for any cell in row 1 with member Row_Member1. The

value for column A is the sum of values in cells outlined in blue ($9+10=19$) when the rule is evaluated for any cell in row 1 with member Row_Member3. Similarly, the value for column A is the sum of values in cells outlined in green ($17+18=35$) when the rule is evaluated for any cell in row 2 with member Row_Member5, and so on.

		A	
		Column_Member 1	Column_Member 2
1	Row_Member 1	1.0	2.0
	Row_Member 2	5.0	6.0
	Row_Member 3	9.0	10.0
2	Row_Member 4	13.0	14.0
	Row_Member 5	17.0	18.0

Row Value

Action:

The action is performed when the value for the specified row satisfies the condition.

Operators:

The operators available for this function: =, !=, <, <=, >, >=, Equals, Not Equals, Contains, Starts With, Ends With. These operators act on the target value selected, which can be a free form value, cell value, column value, row value, or cross-dimension member.

Condition Definition:

IF	Row Value	1	!=	Value	10		
----	-----------	---	----	-------	----	--	--

Condition Evaluation:

A design-time row can expand to one or more data cells at data entry time, as shown in the following figure. The value for a row is the sum of the values in all of the expanded data cells in that row at the current column location. The current column changes as the current cell for which the rule is being evaluated changes within the grid.

For example, the value for row 1 is the sum of the values in the cells outlined in purple ($1+5+9=15$), when the rule is evaluated for any cell in column A with member Column_Member1. Similarly, the value for row 2 is the sum of values in cells outlined in blue ($14+18=32$) when the rule is evaluated for any cell in column A with member Column_Member2, and so on.

		A	
		Column_Member 1	Column_Member 2
1	Row_Member 1	1.0	2.0
	Row_Member 2	5.0	6.0
	Row_Member 3	9.0	10.0
2	Row_Member 4	13.0	14.0
	Row_Member 5	17.0	18.0

Cross Dim Member

Action:

The action is performed when the value in the data cell referenced by the cross-dimension member satisfies the condition. The current data cell members are used to fully qualify the cell for dimensions whose members are not specified in the cross-dimension. This member name is a free form entry.

Operators:

The operators available for this function: =, !=, <, <=, >, >=, Equals, Not Equals, Contains, Starts With, or Ends With. These operators act on the target value selected. The target value can be a free form value, cell value, column value, row value or cross-dimensional member. It can include one member only from each dimension, and must include only members for dimensions on rows or columns.

Condition Definition:

IF	Current Cell Value	>	Cross-Dim Member	Row_Member 5
----	--------------------	---	------------------	--------------

Condition Evaluation:

When the previous rule is applied at grid level, the rule is invoked on each cell in the form, and the value in that cell is compared with the value in the cell outlined in purple. Thus, the cell at Row_Member 5->Column_Member 2 will turn red.

		A	
		Column_Member 1	Column_Member 2
1	Row_Member 1	1.0	2.0
	Row_Member 2	5.0	6.0
	Row_Member 3	9.0	10.0
2	Row_Member 4	13.0	14.0
	Row_Member 5	17.0	18.0

Member Name

Action:

The action is performed if the current data cell on which the rule is invoked has the specified dimension member in its intersection. The member name for the selected dimension should be in the plan type for which the form is created.

Operators:

The operators can be Equals, Not Equals, Contains, Starts With, or Ends With. The target value, which is the member name, is selected or entered free form.

Condition Definition:

IF	Member Name	Account	Equals	Value	Total Cost			
----	-------------	---------	--------	-------	------------	--	--	--

Member

Action:


The action is performed if the current data cell on which the rule is invoked has the specified dimension member (or any one of the members that result from evaluating the specified function) in its intersection.

Operator:

The available operators are In and Not In. The target value, which is the member, is selected or entered free form.

Including Attributes

Rules can include attribute values. If Source Type is **Attribute**, the available operator is Is, and you can type an attribute value directly in the Target Value field. If Source Type is **Member**, and you select the In or Not In operator in the Target Value field, you can select an attribute by

clicking , and then clicking the **Variables** tab of the Member Selection dialog box. You can use the member selection function selector to select functions for the attribute, such as NotEqual and GreaterOrEqual.

When using attribute values in data validation rules, keep in mind the way attributes are evaluated. If a rule references one or more attributes from one or more dimensions, they are evaluated as an OR for attribute values from the same attribute dimension, and as an AND for attributes from different attribute dimensions. For example, if the rule includes attributes IN Red, Blue, True, Big, then all members are selected that are either (Red OR Blue) AND True AND Big. For additional information, see [About Selecting Attribute Values as Members](#).

Condition Definition for Member Source Type

IF	Member	Account	In	Value	IDescendants("Total Cost")			
----	--------	---------	----	-------	----------------------------	--	--	--

Condition Definition for Attribute Source Type

IF	Attribute	Entity	Is	Value	red			
----	-----------	--------	----	-------	-----	--	--	--

Account Type

Action:

The action is performed if the current data cell on which the rule is invoked has an account with the specified account type in its intersection. Refers to all the currently supported Account Types: Expense, Revenue, Asset, Liability, Equity, and Saved Assumption.

Operator:

The available operator is Is.

Condition Definition:

IF	Account Type		Is		Expense			
----	--------------	--	----	--	---------	--	--	--

Version Type

Action:

The action is performed if the current cell on which the rule is invoked has a version with the specified version type in its intersection. It refers to the version types currently supported, standard bottom-up and standard top down.

Operator:

The available operator is Is.

Condition Definition:

IF	Version Type		Is		Standard Botto...			
----	--------------	--	----	--	-------------------	--	--	--

Variance Reporting Type

Action:

The action is performed if the current cell on which the rule is invoked has an account with the specified variance reporting type in its intersection. Refers to the available variance reporting types, Expense and Non-Expense.

Operator:

The available operator is Is.

Condition Definition:

IF	Var Reporting Type	Is	Non-Expense	
----	--------------------	----	-------------	--

UDA

Action:

The action is performed if the current cell on which the rule is invoked has this UDA associated with the specified dimension's member in its intersection. The UDA reference is selected based on the selected dimension. The condition is based on the UDA for this dimension being equal to the selected value. You must select the UDA value from the drop-down list.

Operator:

The available operator is Is.

Condition Definition:

IF	UDA	Scenario	Is	Value	ACTUAL
----	-----	----------	----	-------	--------

Attribute

Action:

The action is performed if the current cell on which the rule is invoked has this attribute associated with the specified dimension's member in its intersection. The attribute reference is selected based on the selected dimension. The condition is based on the attribute for this dimension being the selected target value.

Operator:

The available operator is Is.

Condition Definition:

IF	Attribute	Entity	Is	Value	East
----	-----------	--------	----	-------	------

For additional information about data validation, see:

- [Creating and Updating Data Validation Rules](#)
- [Order of Evaluation and Execution for Data Validation Rules](#)
- [Formatting Cells and Setting the Promotional Path](#)
- [Viewing Data Validation Rules](#)
- [Conditions Supported by the Rule Builder](#)

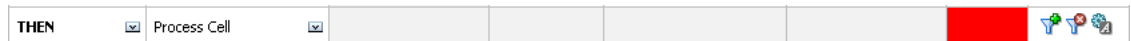
- [Data Validation Conditional Operators](#)
- [Data Validation Rule Scenarios](#)

Then Condition Values

Action:

The Then conditions supported by the data validation rule builder support Process Cell conditions only. To enter Process Cell Conditions, see [Formatting Cells and Setting the Promotional Path](#).

Condition Definition:



For information on other conditions, see [Conditions Supported by the Rule Builder](#).

For additional information about data validation, see:

- [Creating and Updating Data Validation Rules](#)
- [Order of Evaluation and Execution for Data Validation Rules](#)
- [Formatting Cells and Setting the Promotional Path](#)
- [Viewing Data Validation Rules](#)
- [Conditions Supported by the Rule Builder](#)
- [Data Validation Conditional Operators](#)
- [Data Validation Rule Scenarios](#)

Range Condition Values

The Check Range and Range conditions are used together. These conditions can be used in the data validation rule builder in a Then clause or standalone.

See:

- [Check Range](#)
- [Range](#)

For information on other conditions, see [Conditions Supported by the Rule Builder](#).

For additional information about data validation, see:

- [Creating and Updating Data Validation Rules](#)
- [Order of Evaluation and Execution for Data Validation Rules](#)
- [Formatting Cells and Setting the Promotional Path](#)
- [Viewing Data Validation Rules](#)
- [Conditions Supported by the Rule Builder](#)
- [Data Validation Conditional Operators](#)
- [Data Validation Rule Scenarios](#)

Check Range

Action:

Defines the value that needs to be in a specific range.

Value:

This value can be the Current Cell Value or the value in a particular Row, Column or Cell.

Range

Action:

Defines a valid range for the value defined in the Check Range condition. This range includes all values that are \geq the minimum value and $<$ the maximum value. If the value specified in the Check Range condition is within this range, then the processing instructions defined by this condition are applied to the data cell on which the rule is being invoked. You can define multiple ranges of values and provide different processing instructions for each range.

Value:

The minimum and maximum values for the range can be defined using Cell Value, Current Cell Value, Row Value, Column Value, Cross-dimension Value, or by entering a free-form value. For example, the following rule ensures that the current cell value is ≥ 5 and < 10 . If this condition is met, the cell is turned red.

Condition Definition:

Condition	Source Type	Source Value	Operator	Target Type	Target Value	Process	Actions
CHECK RANGE	Current Cell Value						
RANGE	Value	5		Value	10		

For information on other conditions, see [Conditions Supported by the Rule Builder](#).

Data Validation Conditional Operators

Conditional operators in the data validation rule builder can include these types of comparisons:

- Numeric comparisons, using these operators:
 - = (equal to)
 - != (not equal to)
 - < (less than)
 - <= (less than or equal to)
 - > (greater than)
 - >= (greater than or equal to)
- String value comparisons, using these operators:

- Equals
- Not Equals
- Contains
- Starts With
- Ends With
- In
- Not In

Rules can compare cells with different data types; for example, text and Smart List. The data type of the cell is honored if the referenced value always comes from one cell. This is the case when using Current Cell Value and Cross Dim Member to refer to a cell value. In cases where the value being compared comes from multiple cells (such as row value, column value, and cell value), the data type is defaulted to double.

When comparing values for these data types:

- For double, a string representation is used for the double value, such as "123.45." If the double is a whole number with no fractional part, such as 123.00, the integer value is used, for example, "123."
- For Smart Lists, rules use the Smart List name and the numeric value as stored in Oracle Essbase. They do not use the Smart List label because it can change for different users, depending on the user locale.
- For text, rules use only the text value for comparison.
- All other data types (currency, non-currency, percentage, and date) are treated as double.
- For date, rules use the numeric value as stored in Essbase for comparison. For example, if a user types 12/11/1999, assuming the format is MM/DD/YYYY, the value is stored in Essbase as 19991211, and this numeric value is used for comparison.

Table 8-2 Examples of Results for Starts With, Ends With, and Contains

Operator	Compare Value	Compare To Value
Starts With	2.0	2
	1234.0	12.0
	101.0	10
	2.0	2.0
	2.5	"2."
	"YearTotal"	"Year"
Ends With	2.0	2.0
	2.0	2
	2.5	5
	2.5	".5"
	"YearTotal"	"al"
	"YearTotal"	"Total"
Contains	2.0	2.0
	2.0	2
	2.5	5
	2.5	".5"
	2.5	2.5
	23.567	3.5

Table 8-2 (Cont.) Examples of Results for Starts With, Ends With, and Contains

Operator	Compare Value	Compare To Value
	23.567	67
	23.567	"23."
	23.567	".56"
	"YearTotal"	"al"

For additional information about data validation, see:

- [Creating and Updating Data Validation Rules](#)
- [Order of Evaluation and Execution for Data Validation Rules](#)
- [Formatting Cells and Setting the Promotional Path](#)
- [Viewing Data Validation Rules](#)
- [Conditions Supported by the Rule Builder](#)
- [Data Validation Rule Scenarios](#)

Data Validation Rule Scenarios

These scenarios provide examples of how data validation can help implement business policies.

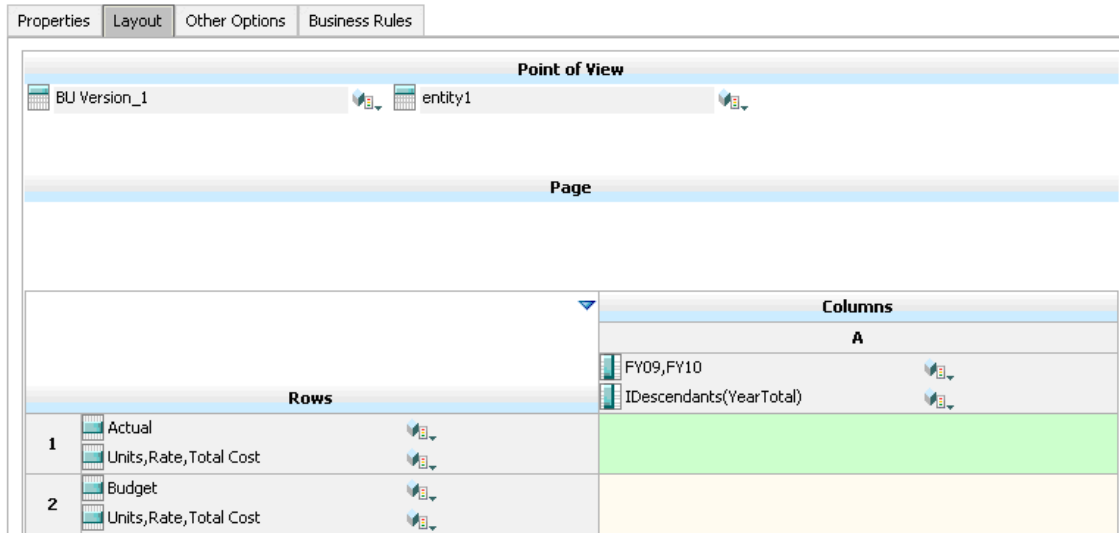
For additional information about data validation, see:

- [Creating and Updating Data Validation Rules](#)
- [Order of Evaluation and Execution for Data Validation Rules](#)
- [Formatting Cells and Setting the Promotional Path](#)
- [Viewing Data Validation Rules](#)
- [Conditions Supported by the Rule Builder](#)
- [Data Validation Conditional Operators](#)

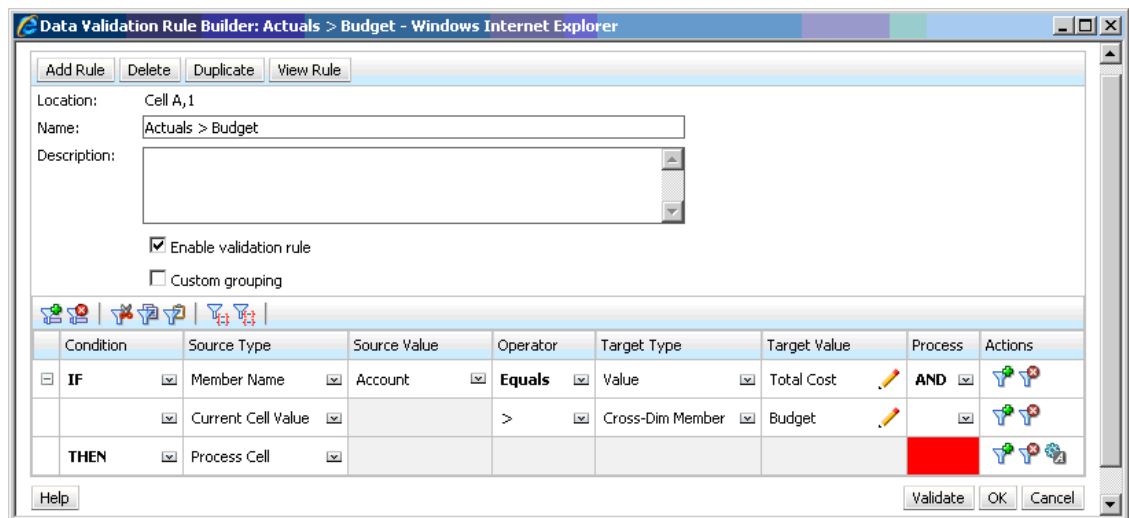
Scenario 1

John is hired by a company called Acme, Inc. as a consultant to design forms and implement data validation rules that enforce some of the company policies. He is asked to implement a validation rule that flags Actual amounts in red if the Total Cost in actuals exceeds the budgeted amount. This test must be repeated for each year and time period in the application. John designs the form and adds a data validation rule at cell level using a cross-dimension member, as shown in the following figures.

Form Layout at Design Time:



Data Validation Rule at Design Time:



Form at Data Entry Time with Data Validations Applied:

		FY09								FY10			
		Jan	Feb	Mar	Q1	Q2	Q3	Q4	YearTotal	Jan	Feb	Mar	Q1
Actual	Units	3	4	6	13	12	24	21	70	5	14	7	26
	Rate	5	5	5	15	15	15	9	54	4	4	4	12
	Total Cost	15	20	30	195	180	360	189	3780	20	56	28	312
Budget	Units	3	4	6	13	12	24	21	70	5	13	7	25
	Rate	4	6	3	13	15	15	9	52	5	4	4	13
	Total Cost	12	24	18	169	180	360	189	3640	25	52	28	325

Tips:

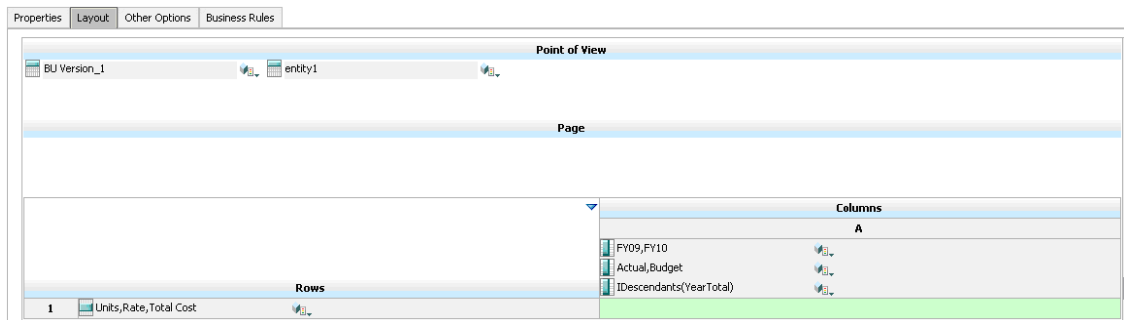
- John can split Total Cost into its own segment and apply the data validation rule at that segment for a slight performance gain. However, doing so would increase maintenance as new accounts and scenarios were added to the form.
- If the requirements changed such that only the YearTotal Period in Actual had to be flagged in red, John would have two options. The best option is to add an IF entry to check if the Period member is YearTotal. Another option is to split the YearTotal member into a separate column for better performance. However, doing so would break the spreading logic, the column header for Year would repeat, and the form would be harder to maintain as new years were added.

For additional scenarios, see [Data Validation Rule Scenarios](#).

Scenario 2

After reviewing the form designed by John in Scenario 1, Acme decides that they want Budget on the column instead of the row. To implement this requirement, John can move members within the axes to change the form layout. However, he does not need to update the data validation rules. John updates the form as shown in the following figure.

Form Layout at Design Time:



Form at Data Entry Time with Data Validations Applied:

	FY09									FY10				
	Actual									Budget	Actual			
	Jan	Feb	Mar	Q1	Q2	Q3	Q4	YearTotal	YearTotal	Jan	Feb	Mar	Q1	
Units	3	4	6	13	12	24	21	70	70	5	14	7	26	
Rate	5	5	5	15	15	15	9	54	52	4	4	4	12	
Total Cost	15	20	30	195	180	360	189	3780	3640	20	56	28	312	

For additional scenarios, see [Data Validation Rule Scenarios](#).

Scenario 3

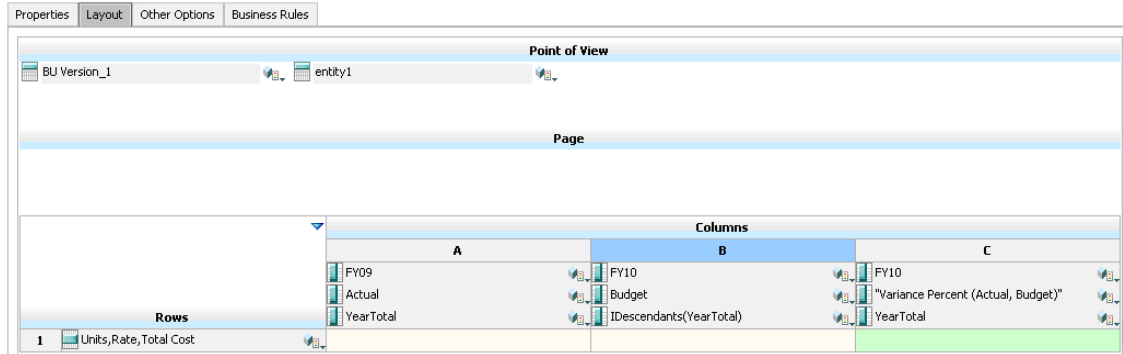
Following the successful rollout of these forms, John is asked to implement the next policy, which is to ensure that this year's Budget amounts are not significantly higher than previous year's Actual amounts. If the difference is greater than 5%, then flag the difference in red.

John decides to use a member with a member formula to calculate the variance between this year's Budget and the previous year's Actual amount. He adds this member formula:

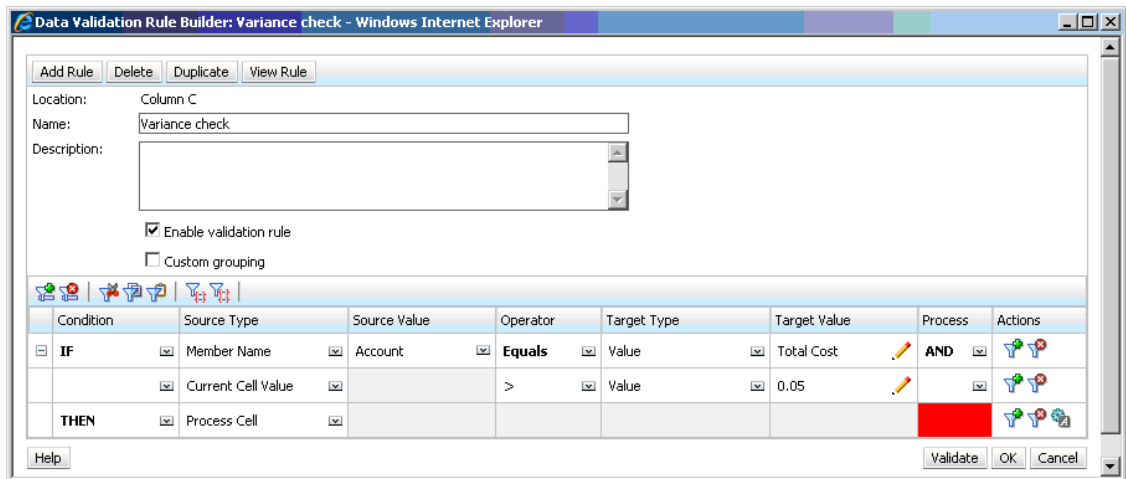
```
@varper(@Prior("Actual", 1, @Relative("Year", 0)), budget)/100;
```

John designs the form and adds a data validation rule at cell level, as shown in the following figure. He uses Member Name to apply the validation only to Total Cost.

Form Layout at Design Time:



Data Validation Rule at Design Time:



Form at Data Entry Time with Data Validations Applied:

Version: BU Version_1							
	FY09	FY10					FY10
	Actual	Budget					Variance Perce
	YearTotal	⊕ Q1	⊕ Q2	⊕ Q3	⊕ Q4	⊖ YearTotal	YearTotal
Units	70.0	60.0	20.0	20.0	15.0	115.0	39.13%
Rate	54.0	24.0	4.0	4.0	5.0	37.0	-45.95%
Total Cost	3780.0	1440.0	80.0	80.0	75.0	4255.0	11.16%

Tips:

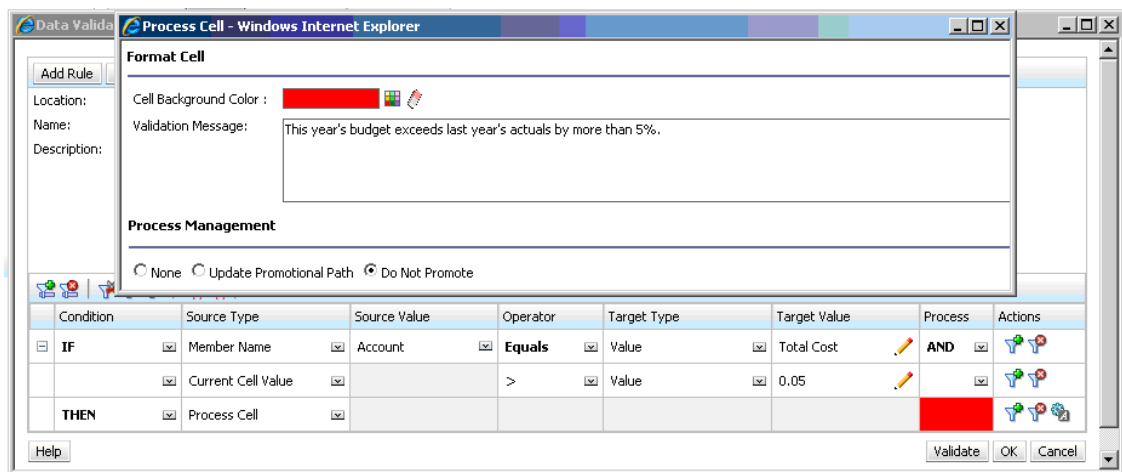
- If John is not allowed to change the outline, or if he experiences performance issues related to member formulas, he can use a formula column. See [Designing Forms with Formula Rows and Columns](#).
- John defines the rule at the Variance Percent column for these reasons.
 - It improves performance. The rule is evaluated only on the cells in the Variance Percent column. If the rule had been assigned to YearTotal, it would have to be evaluated for all Time Periods for the current year budget.
 - It helps users respond to the data validation message. John can add a message to the Variance Percent column stating that the variance is higher instead of adding it to YearTotal. This way, users do not have to look for Variance Percent to determine the difference.
- John could have flagged both YearTotal and Variance Percent in red if this had been part of the requirement.

For additional scenarios, see [Data Validation Rule Scenarios](#).

Scenario 4

In addition to flagging the cell in red, the rule is also required to prevent anyone from promoting the planning unit if this year's Budget is significantly higher (> 5%) than the previous year's Actual amounts. To implement this requirement, all John needs to do is edit the data validation rule's processing instructions and select **Do Not Promote**, as shown in the following figure.

Data Validation Rule at Design Time:



For additional scenarios, see [Data Validation Rule Scenarios](#).

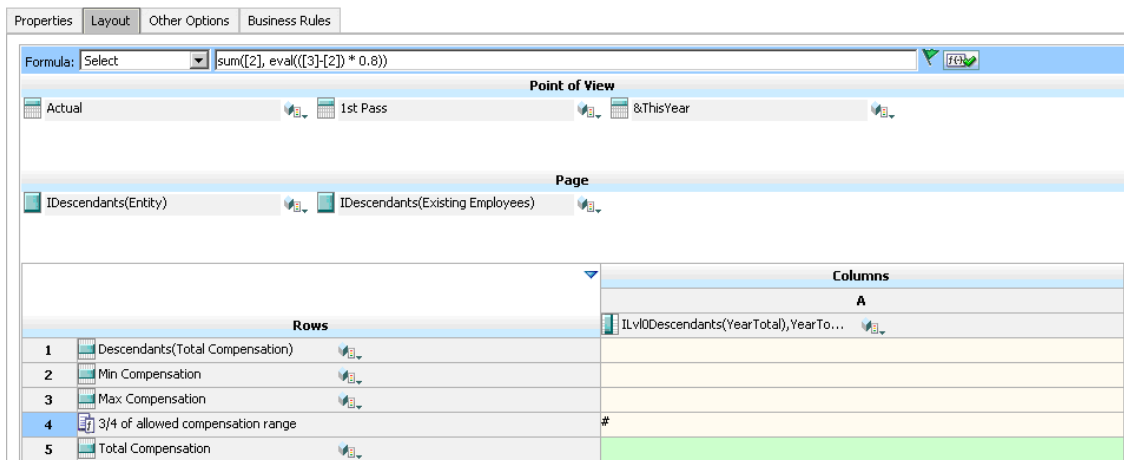
Scenario 5

Finally, John is asked to design a data validation rule to validate that the total compensation for employees in a particular department is within the allowed range. The rule evaluates Existing Employees in the Operations department. It validates that, if Total Compensation is > than Min allowed, and is <= ¾ of the compensation range for the employee's grade, no action is needed.

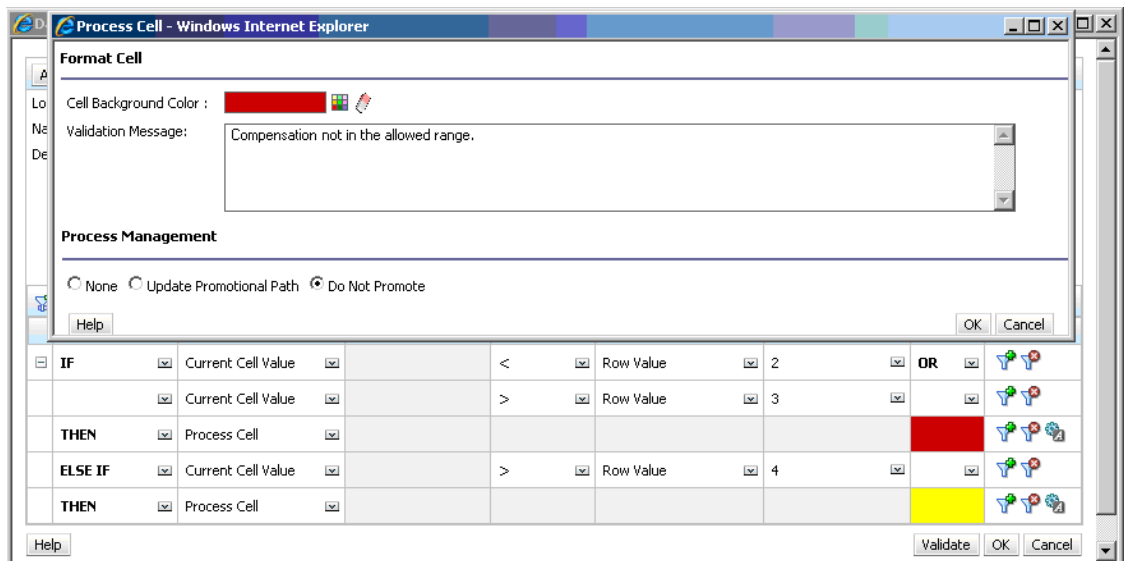
If Total Compensation is greater than $\frac{3}{4}$ of the compensation range, a validation message is provided, and the planning units must be approved by a human resource manager. If the value is less than Min and greater than Max, an error is generated, and users cannot promote their planning units.

John opens the Employee Expenses Summary form in the Form Management dialog box. The form has employees and departments on the page, accounts (such as Total Compensation) on the row, and time period on the column. To make validations easier to build, John adds a calculated row to calculate $\frac{3}{4}$ of the compensation range, and adds Min Compensation and Max Compensation members to the form, as shown in the following figures. Min Compensation and Max Compensation for the employee's grade are calculated using member formulas.

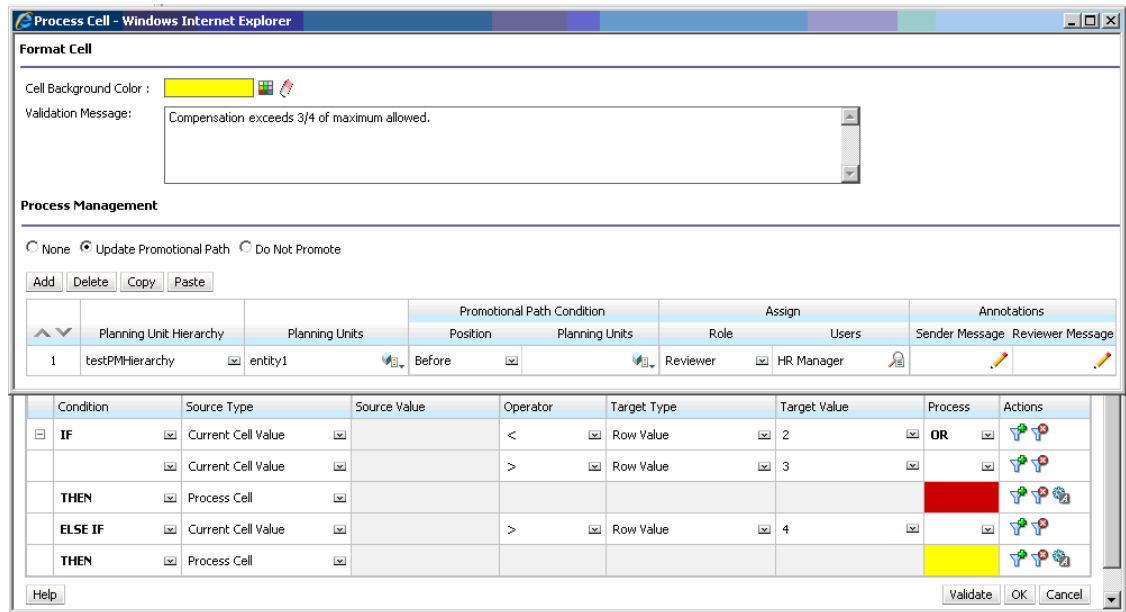
Form Layout at Design Time:



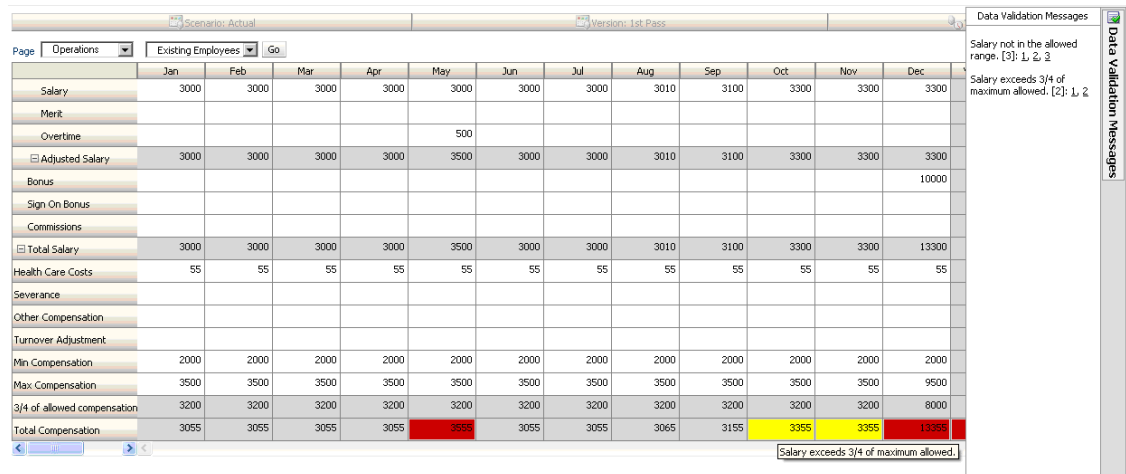
Data Validation Rule to Stop Promotion of Planning Units:



Data Validation Rule to Add the Human Resources Manager as Reviewer:



Form at Data Entry Time with Data Validations Applied and Validation Messages Shown:



For additional scenarios, see [Data Validation Rule Scenarios](#).

9

Managing the Budgeting Process

About the Budgeting Process

You can track budgets and review status, process issues, and planning unit ownership using planning units. Budget cycle time is reduced:

- Approval path is independent of organizational structure
- Exceptions and problem areas are highlighted
- Audit information includes annotations and process status
- Reviews include annotations and comments

Planning Units

Planning units are combinations of scenario, version, and entity or part of an entity. Scenarios and versions are the basis of the review cycle. Planning units submit planning data for a scenario and version. For example, a planning unit might consist of a version (Best Case), an entity (New York), and a scenario (Actual). Planning units can also include secondary dimensions within any entity, refining the granularity of a planning unit.

Review Process

The planning unit moves from one reviewer to another until the budget process is complete. The review process follows the promotional path you set up when you select the owner and reviewers for a planning unit, unless an event triggers a change in the promotional path. Events that affect the promotional path include:

- Exceeding or not reaching expense boundaries for budget items such as salaries, new hires or capital equipment
- The current owner returning the budget to the previous owner for additional information
- The current owner requesting help from an authorized user who is not necessarily on the promotional path

The selected approvals template determines the first user to review the budget (see [Setting Planning Unit Hierarchy Name, Scope, and Template](#)). The first user completes the assigned tasks, then promotes (Bottom Up template) or submits (Distribute template) the budget, which alerts the next owner that the budget requires their attention. Other users may also be notified whenever the budget passes from one user to another.

Each reviewer must validate the planning unit before sending the budget to the next reviewer. The validation runs all data validation rules defined for the planning unit with which the reviewer is working, and reports any data errors or changes in the planning unit promotional path. See [Modifying the Planning Unit Promotional Path](#).

 **Tip:**

To display users' full names (instead of their user IDs) in Approvals, select the option **Display Users' Full Names**, as described in [Specifying System Settings](#).

Planning Unit Hierarchy

A planning unit hierarchy contains planning units and entities that are part of the review process.

Parent/child relationships between planning unit hierarchy members affect the review process:

- When you promote or reject a parent, its children are promoted or rejected unless they are Approved. The owner for the parent becomes the owner of the children.
- When you approve a parent, its children are approved.
- After all children are promoted to the same owner, the parent is promoted to the owner.
- When the status of all children changes to one status, for example Signed Off, parent status changes to the same status.

You cannot change the status of a parent if its children have different owners. If the children are promoted to, submitted to, or signed off by different users, the parent has no owner and only budget administrators can change its status.

Data Validation Rules

To implement business policies and practices, administrators can build data validation rules that are checked when conditions are met in forms. Rules can generate validation messages, enforce limits on submitted planning unit data, and designate a specific reviewer or owner to review data that meets some condition.

For example:

- Conditionalizing the planning unit promotional path
- Preventing the promotion of planning units that include invalid data

When you design rules that affect the planning unit promotional path, you need to understand the order in which these rules are evaluated and applied. For information about designing data validation rules and expected outcomes, see [Managing Data Validation](#).

Task Lists

Task lists guide users through the planning process by listing tasks, instructions, and end dates. Administrators and interactive users create and manage tasks and task lists.

Setting Up Email for Approvals Notification

Administrators must specify the email server details before others can enable email notification. The email server details for approvals notifications come from the EPM Registry, which are set using the EPM System Configurator under Common Settings.

To specify the email server, see the *Oracle Enterprise Performance Management System Security Configuration Guide*.

Approvals notification are enabled on the **Application Settings** page. After email notification is enabled, users receive email when they become owners of planning units, or are specified as a user to notify. This feature is available for SMTP email systems only.

 **Note:**

Approvals email notification must be enabled by the application owner. If they are not enabled by the application owner, then email notifications will not be sent when other users promote or perform other actions on the planning unit, even if the administrator has enabled approvals notifications.

To enable approvals notification, see [Setting Application Defaults](#).

Defining the Budgeting Process

Approvals Roles

To administer approvals for Oracle Hyperion Planning, you must be assigned the appropriate roles.

- **Approvals Administrator**—Approvals Administrators are typically business users in charge of a region in an organization and need to control the approvals process for their region, but do not need to be granted the Planning Administrator role. The Approvals Administrator role comprises the Approvals Ownership Assigner, Approvals Process Designer, and Approvals Supervisor roles of Planning.
- **Approvals Ownership Assigner**—Performs the tasks that a Planner role can perform plus, for any member of the planning unit hierarchy to which they have write access, they can assign owners, assign reviewers, and specify the users to be notified.
- **Approvals Process Designer**—Performs the tasks that can be performed with the Planner role and Approvals Ownership Assigner role, plus, for any member of the planning unit hierarchy to which they have write access, they can change the secondary dimensions and members for the Entities to which they have write access, change the scenario and version assignment for a planning unit hierarchy, and edit data validation rules for forms to which they have access.
- **Approvals Supervisor**—For any member of the planning unit hierarchy to which they have write access they can, stop and start a planning unit and take any action on a planning unit. Approvals Supervisors can perform the preceding actions even if they do not own the planning unit. However, they cannot change any data in a planning unit unless they own it.

For detailed information on these roles, see the *Oracle Enterprise Performance Management System User Security Administration Guide*.

Approvals Process

Oracle Hyperion Planning supports bottom-up, distributed, or free-form budgeting. Typically, high-level users start the planning units containing loaded data, and then delegate data entry into the lowest-level members to their direct reports, who distribute to their direct reports, and so on. Until a budget is distributed, users cannot access it.

To define the budget process, administrators define:

- Planning unit hierarchies
- Owners and reviewers of the planning unit hierarchies
- Validation rules for evaluating submitted data

When a user submits budget data, the data is subjected to validation rules. If the data passes the validations, the budget is promoted to the next owner, and the original user cannot edit the data unless ownership is granted again. The submission process locks the data from being edited by anyone except the current owner.

Approval Operations and Data Validations

During approvals operations that invoke data validations, user variables and context user variables on the forms that are replaced with the distinct union of the result sets of all possible inputs. User variables and context user variables on the POV will be treated as a page.

Note:

To view the results of the user variable substitution, navigate to the data validation report task list. You may need to purposely cause a validation to fail on the form and then run a validation operation for the planning unit for it to display on the report.

Starting and Supporting the Review Process

After administrators start the review process, the planning unit moves from one reviewer to another until the budget process is complete. The selected approvals template determines the first user to review the budget.

To start the review process:

1. Select **Tools**, and then **Manage Approvals**.
2. In **Scenario**, select a scenario.
3. In **Version**, select a version.
4. Click **Go** to display the planning units associated with the selected scenario and version combination.

The planning units listed are enabled for approvals.

If no planning units are started with the selected scenario and version, this message is displayed: **You have not assigned Planning Unit Hierarchy to the selected Scenario and Version combination.**

5. Select **Tree View** and then in **Plan Cycle**, click **Start** to begin the review process.

Note:

If the planning unit hierarchy uses the Bottom Up template, selecting **Start** starts the planning unit and runs the Originate action. These actions set the user defined as the planning unit owner in the planning unit hierarchy as the current owner, and the planning unit status changes to **Under Review**.

6. **Optional:** Select **Exclude** to remove a planning unit from the planning process or from being tracked in the system.



 **Caution:**

After you exclude a planning unit, all associated annotations and history are discarded. Planning unit status is returned to **Not Started** and the owner is set to **No Owner**. Data values are retained.

Printing Planning Unit Annotations

Administrators can check planning unit status by reporting on annotations for a set of scenarios, versions, and planning unit members. Reports can be based on process status. The application name, selected scenarios, versions, and entities, planning unit title, status, author, origin date, and annotations are displayed. Annotation text displays chronologically, with the most recent entry first.

To create and print reports for planning unit annotations:

1. Select **Tools**, and then **Reports**.
2. Select **Planning Unit Annotations**.
3. Under **Planning Units**, select the **Scenario**, **Version**, and **Entity** combination for which you want to generate a report. If **Custom** is selected, click  to select the custom scenarios, versions, and entities.
4. Under **Approvals Status**, select status states.
5. Click **Create Report**.
6. Click **Print**  on the Adobe Acrobat toolbar.

 **Note:**

To ensure that multibyte characters display in reports, see "Multibyte Characters" in the *Oracle Enterprise Performance Management System Installation and Configuration Troubleshooting Guide*. To see the correct language glyph reflected in the reports, you must also make the fonts available in the `java.home` directory. The Oracle Hyperion Planning directory points to the location `EPM_ORACLE_INSTANCE_HOME\common\JRE\Sun\1.6.0\lib\fonts`.

Managing Planning Unit Hierarchies

Use planning unit hierarchies to adapt the budgeting process to all types of organizational requirements.

- Planning units are a combination of the entity and other dimensions. For example, if an application includes all of a company's products, the planning unit hierarchy for North America can include dimensions and members appropriate to products sold in North America. Similarly, the planning unit hierarchy for the European division can include dimensions and members for products sold in Europe. Within the same approvals

hierarchy, Latin America entities can be enhanced using the Account dimension, creating planning units such as Entities by HR, Entities by Capital Expenditures, and Entities by Revenue.

- Use preset budgeting mode templates to create hierarchies that are bottom-up, distributed, or free form.
- Include dynamic links to dimensions based on generation numbers for the entity dimension and the secondary dimension that is used. For example, automatically add generations 0 to 3 in the entity or segment dimension to the planning unit hierarchy. If a change occurs in the dimension, the planning unit hierarchy can be easily updated.
- Import and export planning unit hierarchies.
- Create planning unit hierarchies that differ by scenario and version. For example, the Budget scenario can have a large planning unit hierarchy consisting of departments, accounts, and products, while the Forecast has a simpler process organization with fewer levels of approval.

Creating Planning Unit Hierarchies

Entity is the primary dimension for each planning unit hierarchy. As members are added to the Entity dimension, the inclusion rules you create determine whether a new member is part of the approval process. An administrator can also add members to the hierarchy as exceptions to the planning unit hierarchy structure.

To create a planning unit hierarchy:

1. Select **Administration**, then **Approvals**, and then **Planning Unit Hierarchy**.
2. Click **Create**.

Note:

The approvals dimension is set to Entity. There are no other choices.

3. Create the generic rule that defines which Entity members are included in the approval process (see [Setting Planning Unit Hierarchy Name, Scope, and Template](#)).
4. Select the primary and subhierarchy members to include in the approval process (see [Selecting Planning Unit Hierarchy Members](#)).
5. Assign owners and reviewers for each stage of the approval process and create the planning unit promotional path (see [Assigning Planning Unit Owners and Reviewers](#)).
6. Click **Save**.

Setting Planning Unit Hierarchy Name, Scope, and Template

To set up a planning unit hierarchy:

1. Select **Administration**, then **Approvals**, and then **Planning Unit Hierarchy**.
2. Take an action:
 - To create a new hierarchy, click **Create**.
 - To edit an existing hierarchy, select a planning unit hierarchy, and then click **Edit**.
3. Select **Approvals Dimension**.

4. In **Hierarchy Name**, provide the planning unit hierarchy name.
5. **Optional**: Provide a description.
6. In **Enable Approvals**, select:
 - **All** to add all planning units to the approval process.
 - **None** to include no planning units in the approval process by default.
To add planning unit groups or individual planning units to the approval process, see [Selecting Planning Unit Hierarchy Members](#).
 - **Custom** to define which planning units to include in the approval process.
You can add individual planning units and planning units based on Parent member and generation criteria. See [Selecting Planning Unit Hierarchy Members](#).
7. In **Approvals Template**, select:
 - **Bottom Up** (see [Bottom-Up Budgeting](#)).
 - **Distribute** (see [Distributed Budgeting](#)).
 - **Free Form** (see [Free-Form Budgeting](#)).
8. In **Plan Type**, select the plan type from which aggregated planning unit values are derived.
See [Assigning Planning Unit Owners and Reviewers](#).
9. Take an action:
 - Click **Next** or select **Primary and Subhierarchy Selection** to select the planning unit hierarchy members (see [Selecting Planning Unit Hierarchy Members](#)).
 - Click **Save** and then **Finish** to save changes and close the planning unit hierarchy.

Bottom-Up Budgeting

Data is input at the leaf member level (for example, children of Budget Group) and consolidated by rolling data up the organizational hierarchy. When the budget is started, data is populated for each scenario and user independently. The ownership follows the hierarchy of approval in bottom-up mode. Users can view or edit data based on access permissions defined for the planning unit. The topmost Budget Group owner consolidates individually approved budgets into a final consolidated budget.

Distributed Budgeting

Budget data is entered at the leaf level of the organization, and ownership starts at the top level of the organization. Ownership is then distributed down the organization hierarchy. After ownership reaches the lower levels, budgets are submitted back to the top through the approval process. The top budget group owner reviews, approves, and loads the budgets for budgetary control, transaction control, and reporting.

Free-Form Budgeting



With free-form budgeting, data is input at the leaf member, and planners select the next owner from a drop-down list. The free-form budgeting mode allows planners to select the next owner from a drop-down list. Select this budget template if you are not using the approvals features described in [Creating Planning Unit Hierarchies](#).



Selecting Planning Unit Hierarchy Members

To select planning unit hierarchy members:

1. Take an action:
 - a. From **Approvals Dimension**, click **Next** or select **Primary and Subhierarchy Selection** to continue defining a planning unit hierarchy.
 - b. Select **Administration**, then **Approvals**, and then **Planning Unit Hierarchy**. Select a planning unit hierarchy, then click **Edit**, and then select **Primary and Subhierarchy Selection** to edit members.

Notes:


- At any time during planning unit hierarchy member selection, you can click **Reset to Default Hierarchy** to reset the planning unit hierarchy to its default membership defined in the Approvals Dimension page.
 - Shared members are not displayed in the planning unit hierarchy.
2. Define how to display the planning unit hierarchy:
 - Right-click and select **Expand** or **Expand All Below** to expand the display.
 - Right-click and select **Collapse** or **Collapse All Below** to collapse the display.
 - Select **All Entities** to display all potential planning units.
 - Select **Planning Units** to display only enabled planning units.
 - For **Search**, select **Name**, **Alias**, or **Both**. Enter any part or all of a name in **Search** to locate an entity, then click  to search forward (down) or  to search backward (up) in the planning unit hierarchy.
 - To move from page to page in a multipage planning unit hierarchy, enter a page number in **Page** and click **Go**, or click **Start** (first page), **Prev** (previous page), **Next**, or **End** (last page).
 3. **Optional:** For planning units not included in the default settings for the approval process, check the box to the left of the planning unit name to include the planning unit in the approval process.
 4. **Optional:** Right-click a planning unit name to define subhierarchy members for the approval process, and then select one option:
 - **Include Children** to include the children of the planning unit.
 - **Include Member** to include just the planning unit, but none of its descendants.
 - **Include All Descendants** to include all descendants of the planning unit.
 - **Include Generation** to include one or more planning unit generations. Specify the generations to include when prompted.
 - **Exclude Children** to exclude the children of the planning unit.
 - **Exclude Member** to exclude just the planning unit, but none of its descendants.
 - **Exclude All Descendants** to exclude all descendants of the planning unit.
 - **Exclude Generation** to exclude planning unit generations. Specify the generations to exclude when prompted.
 5. **Optional:** Add a secondary dimension to a planning unit included in the approval process to provide finer granularity:
 - a. Select a dimension from **Dimension**.

- b. In **Parent Member**, click  to display the member selection window, and then select one member as the parent member for that dimension.
 - c. In **Relative Generation**, specify the parent-member generations to include.
Adding a generation adds all members of that generation to the approval process.
 - d. Check **Auto Include** to automatically include newly-added members to the planning unit hierarchy that meet the selected criteria.
 - e. **Optional:** Click  in **Selected Members** for the planning unit to refine the members it includes. Clear the check box next to any member you want to remove.
6. Take an action:
- Click **Next** or select **Assign Owners** to specify planning unit ownership (see [Assigning Planning Unit Owners and Reviewers](#)).
 - Click **Save** and then **Finish** to save changes and close the planning unit hierarchy.

Assigning Planning Unit Owners and Reviewers

Planning unit ownership is inherited from the planning unit parents. Planning unit reviewers are also inherited. You can also explicitly specify planning unit owners and reviewers to assign owners and reviewers other than those planning units inherit.

To assign planning unit reviewers and owners:

1. Take an action:
 - a. From **Primary and Subhierarchy Selection**, click **Next** or select **Assign Owners** to continue defining a planning unit hierarchy.
 - b. Select **Administration**, then **Approvals**, and then **Planning Unit Hierarchy**. Select a planning unit hierarchy, then click **Edit**, and then select **Assign Owners** to specify planning unit ownership.
2. Select a planning unit, then under **Owner**, click  to select an owner.


A planning unit can have only one owner. Either a user or a group can be the owner. Select the **Users** tab to assign an individual user as the owner. Select the **Groups** tab to assign a group as the owner. See [About Group-based Approvals](#).

3. Under **Reviewer**, click  and select planning unit reviewers.

Reviewers can be individual users, a single group, or multiple groups. Select the **Users** tab to assign individual users as reviewers. Select the **Groups** tab to assign a single group or multiple groups as the reviewer. See [About Group-based Approvals](#).

Note:

When using the Bottom Up or Distribute template, if the reviewers are individual users, select the reviewers in the order you want them to review the planning unit. The first reviewer in the list is the first user to work on the planning unit. When the first reviewer promotes or submits the planning unit, the second reviewer selected becomes the planning unit owner, and so on through the list of reviewers you create.



- To enable viewing the aggregated value of planning units during approvals, under **Value Definition Members**, click  to define the intersection of non-planning unit dimensions.

Notes:

- For non-planning unit dimensions:** To view aggregated planning unit values, on **Member Selection**, you must select one member for each non-planning unit dimension in the plan type selected on the **Approvals Dimension** tab. If you do not select members for the non-planning unit dimensions, the aggregated planning unit value is blank.
- For planning unit dimensions:** Value definition members specified for a parent member in the planning unit hierarchy are inherited by the descendants of that member. If value definition members are selected for both a parent and child member of a planning unit dimension, then the child's selection takes precedence. For example, you select a value definition member for both the parent US and its child, Western Region; the value definition member for Western Region takes precedence. At runtime, the Entity, Scenario, Version, and secondary dimension members specified in the planning unit hierarchy are selected.
- In multicurrency applications:** In **Preferences**, users set the **Reporting Currency** in which to display the aggregated planning unit. See the *Oracle Hyperion Planning User's Guide*. To ensure that totals are correct, remind users to launch the Currency Conversion business rule.

 **Note:**

In multicurrency applications, the reporting currency is assumed, so you do not select a value definition member for currency.

- Under **Promotional Path**, click  to display the planning unit promotional path, verify that it is correct, and then correct any errors.
- Under **Notify These Users**, click  to select the users to notify for each move of the planning unit from one user to another.
- Optional:** Repeat Steps 2 through 5 for other planning units to change their inherited owners or reviewers.
- Click **Save** to save your work and continue, or click **Finish** to save your work and close the planning unit hierarchy.

About Group-based Approvals

When assigning Planning unit owners, you can assign individual users, or you can assign a group. When assigning Planning unit reviewers, you can assign individual users, a group, or multiple groups.

Assigning a Group as the Owner

Only one user or one group can be assigned as an owner of a planning unit. Within a group, any user can become the owner, but only one user can be the owner at a time. Only the user assigned as an owner can perform actions. Other group members can take ownership away from the current owner. If no one is assigned as the owner, then anyone in the group can perform actions on behalf of the group without having to first claim ownership.

Assigning a Group or Groups as the Reviewer

If you select individual users as reviewers, all users must approve, and the approvals must follow the order in which the users are entered. If you select a group (or groups) as a reviewer, any user within the group can be the reviewer and can promote to the next level.

The following are some examples of how you can enter multiple reviewers:

Example 9-1 Example 1

```
North America - Bill
    USA - Nick, Sandy, Kim
    CA - John
```

The approval path is John, Nick, Sandy, Kim, Bill.

Example 9-2 Example 2

```
North America - Bill
    USA - {Group A}
    CA - John
```

The approval path is John, any user in Group A, Bill.

Example 9-3 Example 3

```
North America - {Group B}, Bill
    USA - Susan, {Group A}
    CA - John
```

The approval path is John, Susan, any user in Group A, any user in Group B, Bill.

Example 9-4 Example 4

```
North America - {Group B}, Bill
    USA - Susan, {Group A}
    CA - John
```

**A validation rules indicates that if New Hires > 4, then before USA, set {Group HR} as reviewer.





If the condition is true for the CA Planning unit, then the path is: John, any user in Group HR, Susan, any user in Group A, any user in Group B, Bill.

Assigning Planning Unit Hierarchy Scenario and Version Combinations

During the budget process, calculations are run for various scenarios, such as Q1 or FY10. For each scenario, calculations can be run for various versions, for example Initial or Final. Before beginning the budget process, assign planning unit hierarchies to the scenario and version combinations included in the budget process.


To assign planning unit hierarchy scenario and version combinations:

1. Select **Administration**, then **Approvals**, and then **Scenario and Version Assignment**.

2. **Optional:** Click  next to the planning unit to view what scenarios and versions are assigned to it. Click  to close the list.
3. Add a scenario and version assignment.
 - a. Click , in the **Actions** column, for the planning unit.
 - b. Click **Select** in the **Scenario** column, and then select the scenario to associate with the planning unit hierarchy
 - c. Click **Select** in the **Version** column, and then select one or more versions to associate with the selected scenario.
 - d. Click **OK**.
A new assignment row is displayed.
4. **Optional:** Click  to remove a scenario and version assignment.
5. Click **Save** to save the scenario and version assignments and continue or click **Cancel Changes** to undo any changes made since the last save.

Selecting Planning Unit Hierarchy Scenario and Version

To select the planning unit hierarchy scenario and version:

1. Select **Tools**, and then **Manage Approvals**.
2. In **Scenario**, select a scenario.
3. In **Version**, select a version.
4. Click **Go** to display the planning unit hierarchy defined for the selected scenario and version.
5. In **Display**, click **Tree View** to display the planning units as a hierarchy, or click **Flat View** to display the planning units as a list.
6. **Optional:** If you own planning units for this hierarchy, check **Select Mine** to select them.
7. For each planning unit, the following is displayed:
 - **Plan Cycle** (Tree View only), which displays whether the planning unit is started and if it is included in the budget process
 - **Approvals Status**, for example, **1st Pass**
 - **Sub-Status**
 - **Current Owner**
 - **Location**
 - In **Path**, click  to view the potential promotional path
 - In **Actions**, click **Details**, to display planning unit details and add or edit planning unit annotations

Editing Planning Unit Hierarchies

To edit a planning unit hierarchy:

1. Select **Administration**, then **Approvals**, and then **Planning Unit Hierarchy**.

2. Check the planning unit hierarchy with which you want to work.
 3. Click **Edit**.
 4. Select the appropriate tab, depending on the changes you want to make (for example, select **Assign Owners** to edit ownership of the planning units in the hierarchy).
 - **Approvals Dimension**
See [Setting Planning Unit Hierarchy Name, Scope, and Template](#) to modify the planning unit hierarchy description or approvals default scope.
 - **Primary and Subhierarchy Selection**
See [Selecting Planning Unit Hierarchy Members](#) to select or remove members from approvals.
 - **Assign Owners**
See [Assigning Planning Unit Owners and Reviewers](#) to modify planning unit owners or reviewers
 - **Usage**
See [Viewing Planning Unit Hierarchy Usage](#) to view the Oracle Hyperion Planning objects (data validation rules or scenario and version assignments) that reference the planning unit hierarchy
 5. Click **Save** when done.
- To select a planning unit hierarchy scenario and version with which to work, see [Selecting Planning Unit Hierarchy Scenario and Version](#).

Deleting and Renaming Planning Unit Hierarchies

You can delete a planning unit hierarchy if it is not referenced by other Oracle Hyperion Planning objects such as data validation rules or scenario and version assignments. The **Usage** tab displays objects that reference the selected planning unit hierarchy and links you to the objects so you can disassociate them from the hierarchy.

You can also change the name of a planning unit hierarchy. Renaming a planning unit hierarchy does not affect Planning objects that reference the hierarchy.

To delete or rename planning unit hierarchies:

1. Select **Administration**, then **Approvals**, and then **Planning Unit Hierarchy**.
2. Take an action:
 - To delete, select the planning unit hierarchy to delete, and then click **Delete**.
 - To rename, select the planning unit hierarchy to rename, and then click **Rename**. Enter the new name for the planning unit hierarchy in the dialog box.
3. Click **OK**.

If you select a planning unit hierarchy that is referenced by other Planning objects, an error message is displayed. See [Viewing Planning Unit Hierarchy Usage](#) to view and edit the Planning objects that reference the planning unit hierarchy.

Viewing Planning Unit Hierarchy Usage

Planning unit hierarchies might have dependencies such as scenario and version assignments or data validation rules which are defined in forms. If dependencies exist for a planning unit hierarchy, the hierarchy cannot be deleted until the dependencies are removed. The Usage tab

enables you to view planning unit hierarchy dependencies and links you to the forms or scenario and version assignments so you can remove the dependencies, if needed.

To view planning unit hierarchy usage:

1. Select **Administration**, then **Approvals**, and then **Planning Unit Hierarchy**.
2. Select a planning unit hierarchy, then click **Edit**, and then select **Usage** to view planning unit hierarchy dependencies.
3. Select **Forms** to view associated data validation rules:
 - If no data validation rules are associated, a message is displayed.
 - If data validation rules are associated, they are listed by form. Click the rule hyperlink to display the form in edit mode in a new tab. You can then update or delete the data validation rule to disassociate it from the planning unit hierarchy. See [Creating and Updating Data Validation Rules](#).
4. Select **Scenario and Version Assignment** to view associated scenario and version assignments:
 - If no scenario and version assignments are associated, a message is displayed.
 - If scenario and version assignments are associated, they are listed by scenario. Click the version hyperlink to display the assignment in a new tab. You can then remove the scenario and version assignment to disassociate it from the planning unit hierarchy. See [Assigning Planning Unit Hierarchy Scenario and Version Combinations](#).
5. If dependencies are removed during steps 3 or 4, click **Refresh** on the **Usage** tab to view the remaining dependencies.
6. If you are deleting a planning unit hierarchy, repeat steps 3, 4, and 5 until all dependencies are removed.

Once all dependencies are removed from a planning unit hierarchy, you can delete the hierarchy. See [Deleting and Renaming Planning Unit Hierarchies](#).

Synchronizing Planning Unit Hierarchies

When you add, delete, or modify dimension members that are used in planning unit hierarchies, the affected planning unit hierarchy must be synchronized with the changes. When you display the list of planning unit hierarchies, the entry for each planning unit hierarchy specifies whether recent changes are reflected in the planning unit hierarchy. Use this procedure to synchronize dimension member changes with the planning unit hierarchy.

Note:

When you add dimension members, they are added as planning units only if they meet the criteria in the inclusion rules for the planning unit hierarchy. For example, if the added entity is a fourth-generation entity, and the inclusion rules specify generations one through three as planning units, the entity is not added as a planning unit. If the entity is a third-generation member, however, it is added as a planning unit the next time the planning unit hierarchy is edited and saved, or synchronized.

To synchronize changes to planning unit hierarchies:

1. Select **Administration**, then select **Approvals**.

2. Select **Planning Unit Hierarchy**. Under the **Synchronized** column, planning unit hierarchies are labeled as follows:
 - **Synchronized**—Changes are synchronized with the planning unit hierarchy
 - **Not Synchronized**—Changes are not synchronized with the planning unit hierarchy
 - **Dependents Not Synchronized**—Changes are synchronized with the planning unit hierarchy, but not in Oracle Hyperion Public Sector Planning and Budgeting with the associated Decision Package planning unit hierarchies
 - **Locked By user**—A user is editing or synchronizing the planning unit hierarchy

 **Note:**

If a user begins editing or synchronizing a planning unit hierarchy after you display the planning unit hierarchy list, the planning unit hierarchy list does not display "Locked" for the planning unit hierarchy. If you try to synchronize this planning unit hierarchy, the synchronization does not occur, and an error message states that it is being edited.

3. Select a planning unit hierarchy listed as **Not Synchronized** in **Synchronized**, and then click **Synchronize**.

Changes are applied to the planning unit hierarchy, and the list of planning units is updated according to the inclusion rules defined for the planning unit hierarchy.

 **Note:**

You cannot synchronize changes to a planning unit hierarchy that another user is editing or synchronizing.

Exporting Planning Unit Hierarchies

When you export a planning unit hierarchy, you create a file that contains the planning unit hierarchy information. After you create this file, you can copy its contents to an existing planning unit hierarchy (see [Importing Planning Unit Hierarchies](#)).

See [Generating Load Files](#) for information about the file format.

To export planning unit hierarchies:

1. Select **Administration**, then **Approvals**, and then **File Based Import/Export**.
2. Select **Export Planning Unit Hierarchy**.
3. In **Existing Planning Unit Hierarchy Name**, select a planning unit hierarchy to export.
4. Click **OK**.
5. When the **Save** dialog box is displayed, save the export file to a location of your choice.
6. Click **Export** or **Done**. **Export** performs the action, and **Done** closes the dialog box.

Importing Planning Unit Hierarchies

A planning unit hierarchy can be populated with the contents of a planning unit hierarchy import file. The import file is the result of exporting an existing planning unit hierarchy. See [Exporting Planning Unit Hierarchies](#).

Importing the planning unit hierarchy information does not create a planning unit hierarchy. The planning unit hierarchy populated from the export file must exist and have at least a name before the import. Loading planning unit hierarchies first deletes all members of the planning unit hierarchy, and then adds each member specified in the input file as a new member.

 **Note:**

A planning unit hierarchy load deletes an existing member and its children from the hierarchy if the member is not specified in the input file.

To import a planning unit hierarchy:

1. Select **Administration**, then **Approvals**, and then **File Based Import/Export**.
2. Select **Import Planning Unit Hierarchy**.
3. In **Existing Planning Unit Hierarchy Name**, select the planning unit hierarchy receiving the exported information.

 **Note:**

The imported planning unit hierarchy includes the defined owner, reviewers, and rules for determining the promotional path.

4. For **Planning Unit Hierarchy with Ownership**, click **Browse** to select the exported planning unit hierarchy file to import.
5. Click **OK**.
6. Click **Import** or **Done**. **Import** performs the action, and **Done** closes the dialog box.

If the message *Import successful* is displayed, the planning unit hierarchy information in the exported file was successfully copied to the planning unit hierarchy that you selected in **Existing Planning Unit Hierarchy Name**.

If the message *Import not successful. Some items have not been imported* is displayed, click **Details** to view the log file. Correct the errors and retry importing the planning unit hierarchy.

Planning Unit Promotional Path

A planning unit moves from person to person and department to department based on the owners and reviewers you assign to each planning unit and its parents in the planning unit hierarchy.


There are two ways to affect the promotional path of a planning unit hierarchy:



- Modify owners and reviewers for planning units and their descendants using approvals actions (for example, Promote or Reject)
- Use data validation rules to determine whether a change in the planning unit promotional path is necessary


Modifying the Planning Unit Promotional Path

When you select an owner and reviewers for a planning unit and its parents, you set the planning unit promotional path (see [Planning Unit Promotional Path](#)). Sometimes, however, budget calculation results change the person who reviews the budget next. For example, if salaries for a sales group are more than 10% over the previous budget year, approval from someone other than the next reviewer may be required. To automate redirecting the budget in these cases, add conditions and actions to data validation rules that test for these exceptions, and then modify the promotional path when necessary. You can also use data validation rules to stop a planning unit from passing to the next reviewer when data errors exist in the planning unit.

To modify the planning unit promotional path:

1. Create or select a data validation rule (see [Creating and Updating Data Validation Rules](#)).
2. In the **Data Validation Rule Builder**, click  to create promotional path conditions and specify the action taken when a budget calculation does not meet these conditions.
3. Select one action:
 - **Update Promotional Path** to create a promotional path condition that adds reviewers or owners to the promotional path if the condition you set is triggered.
 - **Do Not Promote** to prevent promoting a planning unit to the next reviewer. This action usually indicates invalid data in the budget.
4. In the **Process Cell** dialog box, beneath **Approvals**, click **Add** to add a promotional path condition.

Promotional path conditions are processed in the order they are listed. Use the  and  to move them up or down within the list.

5. **Optional:** To duplicate a promotional path condition:
 - a. Select a promotional path condition, and then click **Copy**.
 - b. Select the promotional path condition that will precede the copied one in the list, and then click **Paste** to add the copied promotional path condition.
6. **Optional:** Click **Delete** to remove a promotional path condition.
7. In **Planning Unit Hierarchy** and **Planning Units**, click  to select the planning unit hierarchy and planning units the promotional path condition affects.
8. In **Promotional Path Condition**, specify the promotional path changes for the planning units selected in Promotional Path Condition. Select:
 - a. In **Position**, specify where the promotional path changes for the planning units selected in **Planning Units**. Select:
 - **Before** to insert the alternate owner or reviewer before ownership reaches the planning units specified in **Promotional Path Condition**.
 - **After** to insert the alternate owner or reviewer after ownership reaches the planning units specified in **Promotional Path Condition**.

- **Before and After** to insert the alternate owner or reviewer before and after ownership reaches the planning units specified in **Promotional Path Condition**.

 **Note:**

After you add promotional path conditions to a data validation rule, the planning unit promotional path displays the potential changes in planning unit ownership as an optional path above the defined promotional path. The display indicates where the path can change and how the alternate path rejoins the defined promotional path.

- b. In the **Planning Units** column of **Promotional Path Condition**, click  to select the planning units the action selected in **Position** affects.


 **Note:**

The selected planning units must be ancestors of the planning units selected in the previous step. If you select planning units that are not ancestors, the promotional path condition is not evaluated when the planning unit hierarchy is validated.

9. In the **Assign** column, specify the role, user type, and the user or UDA for each alternate owner, reviewer, and user to be notified if the rule exception occurs.
 - a. In **Role**, select:
 - **Owner** to select the alternate owner
 - **Reviewer** to select the alternate reviewer
 - **Notified Users** to select the user to be notified
 - b. In **Type**, select:
 - **User Name** to specify the alternate user
 - **From UDA** to select the dimension and to enter the prefix that was assigned to the UDA
 - c. In **Users**, specify:
 - For **User Name**, the name of the alternate user or choose one from the user selector
 - For **From UDA**, the dimension from the **Select a Dimension** drop down, and then specify the prefix that was used in the UDA to indicate that the UDA contains a user name (for example, ProdMgr:)

 **Note:**

Ensure the prefix specified is the same as the prefix that was assigned while creating the UDA, including any spaces. For instance, if there is a space between the prefix and the username (ProdMgr Kim), ensure there is a space after ProdMgr.

10. **Optional:** Provide messages for the users involved in the planning unit promotional path changes:
 - a. Click  in:
 - **Sender Message** to enter the email message text sent to the user promoting the planning unit when the promotional path changes.
 - In **Reviewer Message** to enter the email message text sent to the user receiving the planning unit for review due to the rule exception.
 - b. Click **OK** to save the messages.The text you enter is added to the messages sent when the planning unit changes status.
11. Click **OK** to save the promotional path conditions, and return to data validation rule creation.

Design Considerations for Planning Unit Promotional Path Data Validation Rules

When you design rules that affect the planning unit promotional path, you need to understand the order in which these rules are evaluated and applied. For information about designing data validation rules and expected outcomes, see [Managing Data Validation](#).

Managing Task Lists

Task lists guide users through the planning process by listing tasks, instructions, and end dates. Administrators and interactive users create and manage tasks and task lists.

To manage task lists:

1. Select **Administration**, then **Manage**, then **Task Lists**.
2. Use the **Manage Task Lists** page to configure task list folders and create and manage task lists.

If a form contains promotional path rules, you can view validation reports in a Validation Reports folder.

Working with Task List Folders

To create, rename, move, or delete task list folders:

1. Open the **Manage Task Lists** page.
See [Managing Task Lists](#).
2. In the **Task List Folders** area, perform a task:
 - To create a task list folder, select the folder in which to create the task list folder, and then click **Create**. Enter the name of the task list.
 - To rename a task list folder, select the folder to rename, then select **Actions**, and then select **Rename**. Enter the new name.
 - To move a task list folder, select the folder to move, then above the **Task List Folders** area, click **Move**. Select the destination folder.
 - To delete a task list folder, select an empty folder to delete, and then click **Delete**.

 **Note:**

You cannot move or delete the Task Lists folder. If you try to delete a folder containing additional folders, an error message is displayed.

3. Click **OK**.

Working with Task Lists

Task lists organize groups of tasks for users. You must create task lists before creating tasks.

To create and rename task lists:

1. Open the **Manage Task Lists** page.
See [Managing Task Lists](#).
2. In the **Task List Folders** area, perform a task:
To create a task list:
 - a. Select the folder in which to create the task list.
 - b. Above the **Task List** area, select **Actions**, and then **Create**.
 - c. Enter the task list name.To rename a task list:
 3. Click **OK**.
 4. To define the task list, see:
 - [Adding Instructions to Task Lists](#).
 - [Adding and Defining Tasks](#).

Adding Instructions to Task Lists

To add instructions to task lists:

1. Open the **Manage Task Lists** page.
See [Managing Task Lists](#).
2. In the **Task List Folders** area, select the folder containing the task list to modify, and select the task list.
3. For **Task List**, select the task list to modify, select **Actions**, and then **Edit**.
4. For **Edit Task List**, click **Instructions**.
5. Enter instructions for the task list.
6. Click **Save** and **Close**.

Adding and Defining Tasks

After creating task lists, you can add and define tasks such as entering data in forms and running required business rules. See [Adding Tasks to Task Lists](#)

Adding Tasks to Task Lists

You can set completion dates and alerts for tasks that users perform at runtime. You can also set up email messages, for example, to alert users that a task was not completed by its end date. Alert messages are sent after an "alert date" that you set, and are repeated until the end date is reached for a task. You must configure an email server.

To add tasks to task lists:

1. Open the **Manage Task Lists** page.
See [Managing Task Lists](#).
2. Select a task list, select **Actions**, and then select **Edit Task List**.
3. Select **Actions**, and then **Add Child**.
4. Under **Task Details**, for **Task**, enter a task name.
5. For **Type**, select one:
 - **URL**: Opens a specified URL
 - **Form**: Opens a form.

 **Note:**

You can only select the form associated with the task you are adding. You cannot select the form folder.

- **Business Rule**— Launches a business rule that you specify
 - **Manage Approvals**— Starts the review process with a specified scenario and version
 - **Job Console**— Open the Job Console to view a list of jobs for the current user for the specified job status and job type
 - **Copy Version**— Use Copy Version to copy the current form's data, including supporting detail, annotation, cell text, and cell documents, for the specified source and destination versions
6. Specify the information required for the task type using this table:

Table 9-1 Task Information

Task Type	Action
URL	<p>Perform these tasks:</p> <ul style="list-style-type: none"> In URL, enter a fully qualified URL to associate with this task, such as <code>http://www.company_name.com</code> Select Use Single Sign On to enable users to open a URL for another product that accepts single sign-on. See the <i>Oracle Enterprise Performance Management System Security Configuration Guide</i>). To link to Oracle Hyperion Financial Reporting in the Oracle Hyperion Enterprise Performance Management Workspace, single sign-on is not required. Instead, include the ObjectID to link to (see the <i>Oracle Hyperion Enterprise Performance Management Workspace Administrator's Guide</i>). Go to step 7.
Form	<p>Select the form for users to complete, and then go to step 7.</p> <p>Optional: Select Set Page Member Defaults to select the member from each dimension to display as the default when the task is first opened. After you select this option, you can select the members for the page dimensions. The page member defaults apply until a user updates the form and returns to the task in another session. Where page member defaults are set, they override the most recently used settings in each session.</p>
Business Rule	<p>Perform these tasks:</p> <ul style="list-style-type: none"> From Plan Type, select the plan type associated with the business rule to execute. From Business Rules, select the business rule to execute. Go to step 7.
Manage Approvals	<p>Specify the scenario and version in which the user can start the approvals process, and then go to step 7.</p>
Job Console	<p>Perform these tasks:</p> <ul style="list-style-type: none"> In Job Type select the kind of job, such as to Copy Decision Package, to display in the console. In Status, select the situation of the task such as processing or in error, to display in the console. Go to step 7.
Copy Version	<p>Create a task to copy a version of the current form's data by specifying values for the following, and then go to step 7:</p> <ul style="list-style-type: none"> Scenario Copy From Copy To

7. To enter an end date for the task, select **End Date** and then select:

- a. The month, day, and year. (You can change the date display format in Oracle Hyperion Planning preferences. See [Specifying System Settings](#).)
 - b. The hours, minutes, and AM or PM.
8. To send email message for uncompleted tasks, select **End Date**, and then:
 - a. Select **Repeat Every**, and enter a value.
 - b. Select the frequency for email reminders.
9. To send email messages after the alert date and before the end date, select **Alert**, and then:
 - a. Set the date and time to begin sending messages by selecting the month, day, year, hours, minutes, and AM or PM.
 - b. In the **Alert** area, select **Repeat Every** and enter a value.
 - c. Select the frequency for email reminders.
10. To make task completion dependent on completing a primary task, select **Dependency**.
11. Click **Instructions** to enter information that explains how to complete the task.
12. Click **Save** and then **OK**.

To add tasks to task lists:

Editing Task Lists

You can use the Edit Task List dialog box to update task lists.

Editing Tasks

Use the Edit Task dialog box to modify the type of task, its instructions, the end date and alert date, and email reminder messages.

To edit tasks:

1. Open the **Manage Task Lists** page.
See [Managing Task Lists](#).
2. Select a task list, select **Actions**, and then select **Edit**.
3. Select a task, select **Actions**, and then select **Edit**.
4. In **Edit Task**:
 - Modify the task name.
 - Select another task type. For a description of all available types, see [Adding Tasks to Task Lists](#).
5. For **Instructions**, modify instructions for completing the task.
6. **Optional**: To enter an end date for the task, select **End Date**, and enter the date and time by which users must complete this task.
7. **Optional**: To send email messages if tasks are not completed by the end date, select **End Date**.
 - a. Select **Repeat Every** and enter a value.
 - b. Select the frequency for email reminders.
8. **Optional**: To send email messages after the alert date and until the end date, select **Alert**:

- a. Select the month, day, year, time, and AM or PM.
- b. Select **Repeat Every** and enter a value.
- c. Select the frequency for email reminders.
9. **Optional:** To make completion of this task depend on completing a primary task, select **Dependency**.
10. **Optional:** To edit task properties, click **Property**.
11. Click **Save**.
12. Click **Close**.

Copying and Moving Tasks

To copy or move tasks:

1. Open the **Manage Task Lists** page.
See [Managing Task Lists](#).
2. Perform an action:
To copy tasks:
 - a. Select a task list, then select **Actions**, and then select **Edit**.
 - b. Select the task to copy, and then click **Save As**.
 - c. Enter the name for the new task list.
 - d. Click **OK**.To move tasks:
To cut and paste tasks:
 3. Click **Close**.

Moving and Reordering Task Lists

To move or reorder task lists:

1. Open the **Manage Task Lists** page.
See [Managing Task Lists](#).
2. Perform a task:
To move a task list:
 - a. Select the folder with the task list to move.
 - b. Select a task list, then click **Move**.
 - c. Select the destination folder.
 - d. Click **OK**.To reorder task lists:

Clearing Task Lists

After a task list is completed, you can clear the completion status, end dates, and alerts for all tasks within a selected task list. This allows tasks to be reused for a future planning period.

Alerts are enabled only when task lists contains end dates. Clearing task lists disables alerts by clearing the check boxes for end dates. It does not delete the dates.

To clear task lists:

1. Open the **Manage Task Lists** page.
See [Managing Task Lists](#).
2. Select the folder with the task list to clear, select the task list, and then click **Edit**.
3. From the **Clear** drop-down list, select an option:
 - **Completion Status**: Clears completion status only
 - **End Dates and Alerts**: Clears any alerts that are enabled, based on the end date set for the task
 - **Both**: Clears completion status, end dates, and alerts.
4. Click the arrow to the right of the **Clear** drop-down list.
5. Click **OK**.

Deleting Tasks and Task Lists

To delete tasks and task lists:

1. Open the **Manage Task Lists** page.
See [Managing Task Lists](#).
2. Perform a task:
 - To delete a task, select the folder and task list with the task to delete, and then click **Edit**. Select the tasks to delete, then select **Actions**, and then select **Delete**.
 - To delete a task list, select the folder with the task list to delete, and select the task list. Select **Actions**, and then select **Delete**.
3. Click **OK**.

Linking Tasks to Planning Pages

Use **Copy Link** to link tasks to Oracle Hyperion Planning pages. You can copy and paste the URL address from a page in the Planning application to the instructions for a task. The task list user can then access the Planning page from that location.

To copy a URL to a task:

1. Access the page in the Planning application to which to link a task.
2. Select **Tools**, then **Copy Link**.
3. Select a task list, then select **Edit Task List**.
4. Select a task, then select **Edit Task**.
5. Under **Task - Instructions**, use the browser to paste the URL link. For example, in the Microsoft Internet Explorer menu, select **Edit**, then **Paste**.
6. Click **Save**.
7. Click **OK**.

Assigning Access to Task Lists

You can determine who can view and modify task lists. By default, administrators can manage and assign access permissions for task lists.



Note:

Being assigned to a task list means being able to access and complete tasks in the task list. It does not mean being able to assign tasks to someone else.

Adding Access to Task Lists

To assign access to task lists:

1. Open the **Manage Task Lists** page.
See [Managing Task Lists](#).
2. Select the task list to modify.
3. In the **Task List** area, select **Actions**, and then select **Assign Access**.
4. **Optional:** To migrate a user or group's changed identity or their position in the user directory from Oracle Hyperion Shared Services Console to Oracle Hyperion Planning, click **Migrate Identities**.
5. **Optional:** To remove deprovisioned or deleted users or groups from the Planning database to conserve space, click **Remove Nonprovisioned Users/Groups**.
6. Click **Add Access**.
7. Select the users or groups to access the task list.
 - Click **Users** to display all user names; click **Groups** to display all groups.
 - If there are multiple pages of users and groups, type the page number to go to in **Page**, and click **Go**.
 - Click **Start** or **End** to navigate to the first or last page.
 - Click **Prev** or **Next** to move to the previous or next page.
8. For **Type of Access**, select how users or groups can use the task list:
 - **Assigned:** View and use
 - **Manage:** Modify
 - **Manage and Assign:** View, use, and modify
 - **None:** No access
9. Click **Add**.
10. Click **Close**.

Changing and Removing Access to Task Lists

To change or remove access to task lists:

1. Open the **Manage Task Lists** page.
See [Managing Task Lists](#).
2. Select the folder and task list to modify, then click **Assign Access**.
3. **Optional:** To migrate a user or group's changed identity or their position in the user directory from Oracle Hyperion Shared Services Console to Oracle Hyperion Planning, click **Migrate Identities**.
4. **Optional:** To remove deprovisioned or deleted users or groups from the Planning database to conserve space, click **Remove Nonprovisioned Users/Groups**.
5. Perform a task:
 - To change access to task lists:
 - a. Select users or groups, and click **Edit Access**.
 - b. For **Type of Access**, select:
 - **Assign:** View and use
 - **Manage:** Modify
 - **Manage and Assign:** View, use, and modify
 - **None:** No access
 - c. Click **Set**.
 - To remove access to task lists:
6. Click **Close**.

Importing and Exporting Task Lists

Administrators can use `TaskListDefUtil.cmd` (Windows) or `TaskListDefUtil.sh` (UNIX) to move task list definitions between Oracle Hyperion Planning applications. You can export or import task list definitions to or from an XML file.

`TaskListDefUtil` uses a command line interface and is installed in the `planning1` directory. For the full path to `planning1`, see [About EPM Oracle Instance](#).

To launch the `TaskListDefUtil` utility:

1. Enter the command from the `planning1` directory, using this syntax:

```
TaskListDefUtil [-f:passwordFile] import|exportFILE_NAME|TASK_LIST_NAME|-all
SERVER_NAME USER_NAME APPLICATION
```

Parameter	Purpose	Required?
<code>[-f:passwordFile]</code>	If an encrypted password file is set up, you can use this option as the first parameter in the command line to run the utility with the full file path and name specified in <code>passwordFile</code> . See Suppressing Password Prompts in Planning Utilities .	No
<code>import export</code>	Import or export the task list definition.	Yes

Parameter	Purpose	Required?
<code>FILE_NAME TASK_LIST_NAME -all</code>	When used with <code>import</code> , specify the XML file containing the task list definition. When used with <code>export</code> , specify the task list to export to XML. Use <code>-all</code> with <code>import</code> or <code>export</code> to import or export all XML files or task list definitions in the current application.	Yes (<code>-all</code> is optional)
<code>SERVER_NAME</code>	Server name on which the Planning application resides.	Yes
<code>USER_NAME</code>	Administrator's name.	Yes
<code>APPLICATION</code>	When used with <code>export</code> , the name of the Planning application containing the task list definitions to export. When used with <code>import</code> , the name of the Planning application to which to import the task list definition.	Yes

- If prompted, enter your password.

When you export task list definitions, the utility creates an XML file in the current directory and logs errors in `TaskListDefUtil.log` in the `EPM_ORACLE_INSTANCE/diagnostics/logs/planning` directory. You can copy the utility to any directory and launch it from there to save files to another directory.

Examples:

- To import one file:


```
TaskListDefUtil.cmd import
c:\EPM_ORACLE_INSTANCE\Planning\planning1\TaskList1.xml localhost admin APP1
```
- To export one file:


```
TaskListDefUtil.cmd export TaskList1 localhost admin APP1
```
- To export all task list definitions:


```
TaskListDefUtil.cmd export -all localhost admin APP1
```
- To import all task list definitions:


```
TaskListDefUtil.cmd import -all localhost admin APP1
```

Copying Data

You can copy plans from one dimensional intersection to another, including relational data and supporting detail. For example, you can copy *Budget, FY10, Final* to *Forecast, FY11, First Draft*.

Notes:

- Selected Copy Data settings are preserved for the current session only.
- Copied dimension members must be present in the selected plan types.
- Data must be copied into cells that can accept data. For example, you cannot copy data into read-only or dynamic cells.
- You can copy account annotations, supporting detail, and cell text. You cannot copy planning unit annotations.

- You cannot use this feature with attributes, so do not select attributes to be copied.
- Oracle Essbase data is copied regardless of selections for Copy Data Options.
- Because this is an administrative function, Oracle Hyperion Planning assumes you have full access to data you copy. You are not prevented from copying to planning units that are approved.
- This feature does not calculate data. To perform calculations, such as increasing the forecast by 5%, apply the business rule after copying data.
- For Copy Data to be successful, you must select at least one member for Scenario, Account, Entity, Period, and Version dimensions.

To copy data:

1. Select **Administration**, then **Manage**, and then **Copy Data**.
2. In **Plan Type**, select a plan type, and then click **Go**.
You can copy from one plan type at a time. When you click **Go**, dimensions are displayed for this plan type.
3. For **Static Dimensions**, enter the members for the data intersections:
 - a. In **Dimension**, select a dimension from which to copy.
 - b. In **Members**, click **Member Selection** to make a selection from which to copy. You can select multiple members. You must select at least one member for Scenario, Account, Entity, Period, and Version dimensions.
4. **Optional:** To add another static dimension to the list, click **Add Dimension** and enter dimension members. (To remove a dimension, select **None - Select a Dimension**. The dimension moves to the **Dimensions with Source and Destination** area.)
5. In **Dimensions with Source and Destination**, enter dimensions into which to copy data:
 - a. In **Source**, click **Member Selection**
 - b. In **Destination**, click **Member Selection**.
6. In **Copy Data Options**, select the type of information to copy.
7. Click **Copy Data**.
Data is copied from one intersection to the other. If data is not copied successfully, a message is displayed. You can also check the log file.
8. To copy data for another plan type, select another plan type in step 2, and repeat the procedure.

 **Tip:**

To view the execution status of Copy Data, see "Checking Job Status" in the *Oracle Hyperion Planning User's Guide*.

Clearing Cell Details

You can clear these cell details for a plan type: account annotations, supporting detail, cell text, and cell-level documents. For information on creating and viewing account annotations, supporting detail, cell text, and cell-level documents, see *Oracle Hyperion Planning User's Guide*.


Notes:

- You cannot delete planning unit annotations.
- Because this function is for administrators and interactive users, Oracle Hyperion Planning assumes that you have full access to details you delete.
- Oracle recommends that you back up the application before performing this procedure. See the *Oracle Enterprise Performance Management System Backup and Recovery Guide*.
- You can also clear cell details with SQL scripts. See [Deleting Application Information Using SQL](#).

To clear cell details:

1. Select **Administration**, then **Manage**, the **Clear Cell Details**.
2. For **Plan Type**, select a plan type, and click **Go**.

When you click **Go**, dimensions are selectable for this plan type.

3. Select members for the data intersections:
 - a. In **Dimension**, select at least one dimension with details to delete.
 - b. For displayed dimensions, click . On the **Member Selection** page, make a selection that includes the details to delete.

 **Note:**

For every dimension selected, you must select at least one member. If a dimension is not selected, Planning includes all its members when clearing cell details.

 **Caution:**

To clear exactly the data you want, select at least one Account member and members from all other dimensions. If you do not select at least one Account member, no data is cleared, even if you select the **Yes, set value to #Missing** option. If you do not select members from the other dimensions, all data for the selected Account member is cleared across all other dimensions.

4. **Optional:** Further refine the data intersection by specifying more members:
 - To select another dimension so you can select its members, click **Add Dimension**.
 - To select all dimensions in the plan type, click **Add All Dimensions**.Select members for the displayed dimensions.
5. Specify the type of information to delete by selecting at least one option from **Clear Options**.
6. Click **Clear**.
A Confirmation page displays your selections.
7. Click **Finish** to proceed, or **Back** to change your selections.

If Clear Cell Details is successful, data is deleted from the plan type. If data is not deleted successfully, a message is displayed. You can also check the log file.

8. **Optional:** To view the execution status of Clear Cell Details and review the information that was deleted, select **Tools**, then **Job Console**.

See "Checking Job Status" in the *Oracle Hyperion Planning User's Guide*.

9. To clear cell details for another plan type, select another plan type in step 2, and repeat the procedure.

10

Working with an Application

Managing Forms and Folders

Use the Form Management and the Business Rule Folders pages to manage folders and forms.

Task	Topic
Create folders	See Creating Folders
Move folders	See Working with Folders
Delete folders	See Working with Folders
Rename folders	See Working with Folders
Create forms	See Creating Simple Forms
Assign access to forms and folders	See Assigning Access to Forms and Folders
Move forms	See Moving, Deleting, and Renaming Forms
Delete forms	See Moving, Deleting, and Renaming Forms

To view all forms or business rules in a Oracle Hyperion Calculation Manager folder, click the folder's name in the lefthand folders area. To select all the forms, select the check box at the top of the forms list.

Creating Folders

Use folders to hierarchically organize forms and Oracle Hyperion Calculation Manager business rule security. You can move folders within the hierarchy, and give folders the same name if they are on different hierarchical levels. You cannot:

- Delete folders unless they are empty
- Select multiple folders
- Rename, move, or delete the top-level folder called, respectively, **Forms** and **CalcMgrRules**

To create folders:

1. Perform an action:
 - For form folders: Select **Administration**, then **Manage**, and then **Forms and Ad Hoc Grids**.
 - For Calculation Manager business rule folders: Select **Administration**, then **Manage**, and then **Business Rule Security**.
2. Select the folder under which to create the folder.
3. Above the folders list, click **Create**.
4. Enter the folder name.

5. Click **OK**.

Working with Folders

To move, delete, or rename folders:

1. For form folders: Select **Administration**, then **Manage**, then **Forms and Ad Hoc Grids**.
For business rule folders: Select **Administration**, then **Business Rule Security**.
2. Select the folder to move, delete, or rename.
3. Perform a task:
 - To move, click **Move**. Select the destination folder to which to move the selected folder.

 **Note:**

When you move folders, all nested folders, forms, and Oracle Hyperion Calculation Manager business rules within them are also moved.

- To delete, click **Delete**.
 - To rename, click **Rename**, and then enter the new name.
4. Click **OK**.

About Setting Preferences

On the Preferences page, all users can set individual preferences. Administrators and application owners can specify global settings. Preference selections affect only the current application.

Preference options depend on user type. Planners and interactive user types can access Application Settings and Display Options. See *Oracle Hyperion Planning User's Guide*.

See:

- [Setting Personal Preferences](#)
- [Setting Application Defaults](#)
- [Specifying System Settings](#)
- [Limiting Use of an Application During Maintenance](#)
- [Specifying Custom Tools](#)
- [Setting Display Options](#)
- [Setting Printing Options](#)

Administrators can control which tabs are displayed for setting personal preferences, application defaults, and system settings.

Show Option	Description
Current Application Defaults	Accesses the Application Settings tab and Display Options tab. The values set on these tabs become application defaults. Users can override defaults, and can revert to defaults by selecting Use Application Defaults on preference tabs. See <i>Oracle Hyperion Planning User's Guide</i> .
Advanced Settings	Accesses the System Settings tab. See Specifying System Settings . Access the Custom Tools tab. See Specifying Custom Tools .

Setting Personal Preferences

All users can set personal preferences in the current application. For Application settings, Display Options, and Printing Options, see *Oracle Hyperion Planning User's Guide*.

To set personal preferences:

1. Select **File**, and then **Preferences**.
2. Click **Planning**, and then perform an action:
 - Select **Application Settings** to set email options, select an alias table, and set options for member selection and approvals.
 - Select **Display Options** to set options for number formatting, page selection, warnings for large forms, and the number of dimensions to show on a page.
 - Select **Printing Options** to specify how pages are printed.
 - Select **User Variables Options** to limit the number of members that display on a form by setting a user variable. See [Managing User Variables](#).

Selecting **Use Application Default** resets the value to the current application default.

Setting Application Defaults

Administrators can specify defaults for the current application. Although users can override these application defaults by setting preferences, they can restore application defaults by selecting **Use Application Default** where available. For information about the individual application settings and display option defaults, see *Oracle Hyperion Planning User's Guide*.

To set application defaults:

1. Select **Administration**, then **Application**, then **Settings**.
2. Select **Current Application Defaults**.
3. Perform one action:
 - Select **Application Settings** to specify defaults for:
 - Email notifications for task lists, approvals, and job console

 **Note:**

Approvals email notification must be enabled by the application creator. If they are not enabled by the application creator, then email notifications will not be sent when other users promote or perform other actions on the planning unit, even if the administrator has enabled approvals notifications.

- Copy the application owner on emails
 - The alias table used
 - The kind of member and alias data to display, such as both name and alias, on the Member Selector dialog box
 - Show planning units as aliases
 - Show planning units that are not started
 - The attribute dimension date format.
 - Select **Display Options** to specify defaults for:
 - Number formatting
 - Remember selected page members
 - Allow search when pages exceed a certain number
 - Member indentation
 - The number of grid rows and columns fetched to populate forms
 - UI themes
 - Date formats
4. Click **Save**.

Specifying System Settings

Only administrators can specify system-wide settings.

Only the application owner can set the Email Server and Password for Synchronizing With Oracle Essbase.

To specify system settings:

1. Select **Administration**, then **Application**, then **Settings**.
2. Select **Advanced Settings**.
3. Select **System Settings**.
4. Set options:

Table 10-1 System Settings

Option	Description
Email Character Set	The character set for email messages: <ul style="list-style-type: none"> • UTF-8 • Regional Setting

Table 10-1 (Cont.) System Settings

Option	Description
Business Rules Notification	Set to Yes to notify users or groups when rules (which are enabled for notification in Oracle Hyperion Calculation Manager) are completed or encounter errors.
Notify These Users	If Business Rules Notification is enabled, select one user or one group to notify.
Shared Services URL	The URL for the Oracle Hyperion Shared Services server. Click Register Shared Services and assign the application to an application group (see Assigning Applications to Shared Services Application Groups).
Enable Use of the Application for	Determine whether users can access the application in maintenance mode, such as during backups. See Limiting Use of an Application During Maintenance .
Enable the Display of Substitution Variables	Set how substitution variables display in the Member Selection dialog box when users respond to runtime prompts in business rules: <ul style="list-style-type: none"> • Display All: Display all substitution variables • Display None: Do not display substitution variables • Enable Filtering: Display only substitution variables that are valid for the runtime prompt
Select User	bug Assign ownership of the application to another administrator.
Display Users' Full Names	Yes: Show full names (such as Victoria Hennings). No: Do not display full names. Only user IDs will be shown (such as VHennings).
Calculation Module	The module used to create, validate, deploy, and administer business rules. (Calculation Manager is the only option.)
Select User	Assign an administrator as the application owner. (By default, the person who creates the application is the application owner. The owner may grant ownership to another administrator.)

5. To specify system settings for each application, repeat these steps, then click **Save**.

Assigning Applications to Shared Services Application Groups

To assign a Oracle Hyperion Planning application to a Oracle Hyperion Shared Services application group:

1. Select **Administration**, then **Application**, then **Settings**.
2. For **Show**, select **Advanced Settings**.
3. Click **Go**.
4. Select **System Settings**.

5. Click **Register Shared Services**, then select an option for **Assign Application to Application Group**:
 - **New Application Group**. Enter the application group name in the text box (available only if you have the Project Manager role in Shared Services).
 - **Existing Application Groups**. Select the application group to which to assign the current application.
6. Click **Submit** and **Save**.

Limiting Use of an Application During Maintenance

Administrators can withdraw and grant access to an application during maintenance. If users are logged on to the application and administrators withdraw their access, users are forced off the system.

To limit use of an application:

1. Select **Administration**, then **Application**, and then **Settings**.
2. Select **Advanced Settings**, and then select **System Settings**.
3. For **Application Maintenance Mode**, select an option for **Enable Use of the Application for**:
 - **All users**: All users who have access to the application can log on or continue working with the application.
 - **Administrators**: Only administrators can log on. Other users are forced off and prevented from logging on until the option is reset to **All Users**.
 - **Owner**: Only the application owner can log on. All other users are prevented from logging on. If they are currently logged on, they are forced off the system until the option is reset to **All Users** or **Administrators**. Only the application owner can restrict other administrators from using the application.
4. If your selection is more restrictive than the current setting, click **OK**.
5. Click **Save**.

See also [Using the MaintenanceMode Utility to Limit Application Access](#).

Using the MaintenanceMode Utility to Limit Application Access

Administrators can use the utility `MaintenanceMode.cmd` (Windows) or `MaintenanceMode.sh` (UNIX) to temporarily withdraw access to applications during maintenance.

To use the `MaintenanceMode` utility:

1. At the command line, navigate to the `planning1` directory.
For the full path to `planning1`, see [About EPM Oracle Instance](#).

2. Enter this command, one space, and the parameters, each separated by a space:

```
MaintenanceMode [-f:passwordFile], /A=app, /U=user, /P=password,

/LL=loginLevel [ALL_USERS|ADMINISTRATORS|OWNER], [/DEBUG=[true|false]],

[/HELP=Y]
```

Table 10-2 MaintenanceMode Utility Parameters

Parameter	Description	Required?
<code>[-f:passwordFile]</code>	Optional: If an encrypted password file is set up, use as the first parameter in the command line to read the password from the full file path and name specified in <i>passwordFile</i> . See Suppressing Password Prompts in Planning Utilities .	No
<code>/A=app</code>	Application name	Yes
<code>/U=user</code>	Name of the administrator executing the utility	Yes
<code>/P=password</code>	The administrator's password	Yes
<code>/LL=loginLevel [ALL_USERS ADMINISTRATORS OWNER]</code>	Specify which users the utility affects: ALL_USERS - All users can log on or continue working with the application. ADMINISTRATORS - Only other administrators can log on. Other users are forced off and prevented from logging on until the parameter is reset to ALL_USERS. OWNER - Only the application owner can log on. All other users are prevented from logging on. If they are currently logged on, they are forced off the system until the option is reset to ALL_USERS or ADMINISTRATORS. Only the application owner can restrict other administrators from using the application.	Yes
<code>/DEBUG=[true false]</code>	Specify whether to run the utility in debug mode. The default is <code>false</code> .	No
<code>/HELP=Y</code>	View the utility syntax online.	No

For example, on a Windows system, to force all users except administrators off an application named "planapp", enter:

```
MaintenanceMode.cmd /A=planapp, /U=admin, /P=password,

/LL=ADMINISTRATORS
```

Specifying Custom Tools

Administrators can specify custom tools, or links, for users on the **Tools** page. Users having access to links can click links from the **Tools** menu to open pages in secondary browser windows.

To specify custom tools:

1. Select **Administration**, then **Application**, and then **Settings**.
2. For **Show**, select **Advanced Settings**.
3. Click **Go**.
4. Select **Custom Tools**.
5. For each link:
 - For **Name**, enter the displayed link name.
 - For **URL**, enter a fully qualified URL, including the `http://` prefix
 - For **User Type**, select which users can access the link.
6. Click **Save**.

Setting Display Options

Administrators can set the number of items that display on the dimensions page and Add Access pages.

To set the number of items that display:

1. Select **File**, then **Preferences**, and then **Display Options**.
2. Enter values:
 - **Show the Specified Members on Each Dimensions Page**
 - **Show the Specified Records on Each Assign Access Page**
3. Click **Save**.

Setting Printing Options

To set printing options, see [Setting Form Precision and Other Options](#).

Mapping an Application for Reporting

Related Topics

- [About the Reporting Application](#)
- [Creating a Reporting Application](#)
- [Mapping a Planning Application to a Reporting Application](#)
- [Defining Application Mappings](#)
- [Defining Dimension Mappings](#)
- [Setting the Point of View](#)
- [Setting Data Options](#)

- [Validation Rules for Default Members](#)
- [Pushing Data](#)
- [Unsupported Features with Mapping Applications](#)
- [Mapping Applications and Substitution Variables](#)
- [Adding an Essbase Server for Reporting Applications](#)
- [Editing an Essbase Server for Reporting Applications](#)
- [Removing an Essbase Server for Reporting Applications](#)

About the Reporting Application

A reporting application is a target application on which you can report on and aggregate Oracle Hyperion Planning data. The primary use cases:

- The reporting application contains data from any source, such as a data warehouse. You want to push new Planning data to it and report on the data.
- You want to report on Smart Lists in Planning, which you cannot do in Planning.

You can map dimensions between a source Planning application and a reporting application to enable:

- Reporting on Planning data in a reporting application
- Aggregating and querying on Smart Lists, which are converted to regular dimensions in the reporting application
- Linking Planning data to multiple reporting applications for various consolidations

Creating a Reporting Application

To create a reporting application:

1. If it does not exist, create the target reporting application.

A reporting application can be either block storage or aggregate storage.

For the characteristics of each, see the *Oracle Essbase Database Administrator's Guide*.

2. In the reporting application, create the dimensions on which you want to report or aggregate.

When you later map the application, Oracle Hyperion Planning automatically maps same-named dimensions. To report or query on Smart Lists, create dimensions in the reporting application that map to the Smart Lists in the source Planning application.

3. Into the reporting application, load members on which you want to report or aggregate.

See [Importing and Exporting Data and Metadata](#).

Next:

- Map the source Planning application to the target reporting application.
See [Mapping a Planning Application to a Reporting Application](#).
- Push data from the source to the target application.
See [Pushing Data to a Reporting Application](#).
- Report on the data in the reporting application.

Mapping a Planning Application to a Reporting Application

To map a Oracle Hyperion Planning application to a reporting application:

1. Create the reporting application.
See [Creating a Reporting Application](#).
2. In Planning, select **Administration**, then **Map Reporting Application**.
3. On the **Reporting Application Mapping** page, select an option:
 - To create a mapping, click **New**.
 - To update a mapping, select the mapping and click **Edit**, **Delete**, or **Rename**. See [Defining Application Mappings](#).
 - To refresh dimensions and members from the reporting application, click **Refresh**. When you are defining an application mapping, Refresh ensures that the dimensionality in the member selector for the target application reflects its actual dimensionality.
 - To push the data to the reporting application, click **Push Data**. See [Pushing Data to a Reporting Application](#).
 - To update the mapping for Smart Lists, see [Synchronizing Smart Lists in Reporting Applications](#).

Defining Application Mappings

To define an application mapping:

1. **For new mappings:** Enter a name and description.
2. Under **Source Application**, select a **Plan Type** from the available plan types for the current application.
The plan type drives the information that is pushed to the reporting application.
3. Under **Reporting Application**, select the Oracle Essbase server on which the reporting application resides, and then select the target reporting application.
4. **Optional:** To add, edit, or remove displayed Essbase servers, see [Adding an Essbase Server for Reporting Applications](#), [Editing an Essbase Server for Reporting Applications](#), or [Removing an Essbase Server for Reporting Applications](#).
5. Click **Next**.
See [Defining Dimension Mappings](#).

Defining Dimension Mappings

On the **Map Dimensions** tab, you map the Oracle Hyperion Planning dimensions on the left to the reporting application members on the right. If mappings are correct, data can be pushed when dimensions in both applications are mapped or have valid default members for storing data in the reporting application (see [Setting the Point of View](#)).

To define dimension mappings:

1. For each Planning dimension, select a **Mapping Type**:

- **Dimension to dimension:** Displays the available unmapped dimensions in the Planning application. In order to push data, the dimension and all of its members must reside in both the source and target applications.

Dimensions that are identical between the Planning and reporting applications (for example, Scenario to Scenario) are automatically mapped.

- **Smart List to Dimension:** Displays the available Smart Lists. When a Smart List is selected, the account members associated with it are displayed. If there is only one member, it is automatically selected.

 **Note:**


Smart List labels are used for mapping to member names or member aliases in the reporting application.

2. In **Dimension/Smart List name**, select the name of the dimension or Smart List.

With Smart List to dimension mapping, the source plan type must contain a dense Account dimension with at least one member associated with a Smart List.

 **Note:**

When mapping Smart Lists to dimensions, you can synchronize Smart Lists to identify dimensions in reporting applications to which Smart Lists are mapped. This also adds level 0 members of the selected dimensions as new Smart List entries to the selected Smart Lists. See [Synchronizing Smart Lists in Reporting Applications](#).

3. In **Member Selection**, select the member name by clicking .

By default, `Lev0Descendants` is selected. You can select only level 0 members. If there is only one member, this member is automatically selected.


4. Click either:
 - **Next** to display the Point of View. See [Setting the Point of View](#).
 - **Save** if all dimensions are mapped and there is no need for a POV. (If so, the **Next** button is grayed out.)

Setting the Point of View


The Point of View tab displays, for each application, dimensions that are not already mapped or that are missing from either application so that you can specify a POV. All POV dimensions must have only one member selected.

To set the point of view:

1. On the top section of the **Point of View** page, specify a member for each unmapped Oracle Hyperion Planning dimension. Either:
 - Enter the member name.

- To select members that are not displayed, click , and then select the POV member.

For rules on valid default members, see [Validation Rules for Default Members](#).

2. In the bottom section, select a member for each unmapped reporting application POV member. Either:
 - Enter the POV member name.
 - To select members that are not displayed, click , and then select the POV member.
3. Click **Save**.

Setting Data Options

Use the Data Options tab to define how to merge relational data such as comments, attachments, and supporting detail when pushing data.

To set data options:

1. Select an option:
 - **Do not copy**—Do not copy relational data.
 - **Ignore when multiple cells merge into one cell**—Do copy relational data if multiple cells form one cell.
 - **Overwrite when multiple cells merge into one cell**—Replace existing cells and attachments with the latest supporting detail data when cells merge
 - **Append multiple cell data into one cell**—If multiple cells merge into one cell, add, append or merge the supporting detail from all the source cells into the target cell.
2. Click **Save**, and then **Finish**.

Validation Rules for Default Members


Default members in the reporting application store the data that is pushed from the source Oracle Hyperion Planning application. If any of the following constraints are not met, or if a dimension in either application is not mapped and has no valid default member, then a mapping is not valid and an error message is displayed.

Rules:

- If the reporting application is an aggregation storage database, then the default members must be level 0 members.
For information on block versus aggregate storage application databases, see the *Oracle Essbase Database Administrator's Guide*.
- If the reporting application is a block storage database, then the default members can be any members with the Store Data property.
- If the Planning application has only dimension to dimension mappings, then the default members can be any level or data storage type.
- If the Planning application has Smart List to dimension mappings, then default members must be only level 0. In addition, the source plan type must contain a dense Account dimension with at least one member associated with a Smart List.

- If Descendants (Acct_Default) is selected in a mapping, the Acct_Default member must exist in the reporting application.

 **Note:**

Mappings that were once valid can become invalid if dimensions, members, or Smart Lists are renamed, removed, or added. If a target plan type has a change in dimensionality, you must select the corresponding application mapping on the **Map Reporting Application** screen and click  to refresh the data.

Pushing Data

Pushing Data to a Reporting Application

After setting up application mappings, you can push data to a reporting application. Oracle Hyperion Planning validates the selected application mappings, and then pushes the mapped Planning dimension data to the reporting application dimensions. You can also check the Job Console for the job status. See [Mapping a Planning Application to a Reporting Application](#).

 **Note:**

If you enabled parent members for dynamic children in both a block storage and aggregate store application and added a dynamic member, you can use **Push Data** to successfully push data from the block storage to the aggregate storage application without refreshing the database.

 **Note:**

If the aggregate storage application is created outside of Planning with members that match the names of the dynamic members used in Planning, pushing data will fail because the system looks for the Oracle Essbase bucket name. This problem does not occur if the aggregate storage database is created within Planning.

To push data to a reporting application:

1. Create the reporting application.
See [Creating a Reporting Application](#).
2. In Planning, select **Administration**, and then **Map Reporting Application**.
3. On the **Reporting Application Mapping** page, click **Push Data**.
4. Select an option:
 - **Clear data on destination and push data:** Clears the data in the target reporting application before pushing data to it.
When the target is an aggregate storage reporting application, note the following when using the **Clear data on destination and push data** option:

- Members with nonmatching names in the target reporting application are ignored.
- This option works only with member names, not with member aliases.
- Use caution when using member relationships (such as **Children**) when selecting members for the application mapping, because using this option can cause the calculation script to exceed its length limit.
- If you use member relationships, this option expands the level 0 member list in the source Planning application. If at least one member name in the source application matches a member in the reporting application, this option proceeds without error. If at least one member does not match, the option cannot proceed.

When the target is a block storage reporting application, to proceed successfully, the **Clear data on destination and push data** option requires these conditions:

- If you use member relationships, all member names in the source application must match all member names in the reporting application.
- If you map Smart Lists to dimensions, all Smart List entries in the source application must match all member names in the reporting application.
- If you map Smart Lists to dimensions, the Smart List entry label in the source Planning application must match the member name in the reporting application. If a Smart List entry label does not match a member name in the reporting application, then the Smart List entry name must match the reporting application member name.

If the previous conditions are not met, the **Clear data on destination and push data** option cannot proceed.

- **Push data:** Pushes the data to the reporting application without first clearing the data in the target reporting application.

If you map Smart Lists to dimensions, Smart List labels must match either member names or aliases in the reporting application. **Push data** does not work with Smart List entry names.

5. Click **OK**.

Pushing Data with a Utility

Pushing data to a reporting application can take some time, so administrators may want to schedule the data push during non-peak hours using the `PushData` utility. The utility is installed in the `planning1` directory. For the full path to `planning1`, see [About EPM Oracle Instance](#).

To launch the `PushData` utility:

1. From the `planning1` directory on the server where Oracle Hyperion Planning is installed, enter this syntax at the Command Prompt:

```
PushData [-f:passwordFile] /U:username /A:sourceApplication /M:applicationMapping
[/C]
```

Parameter	Description
<code>[-f:passwordFile]</code>	Optional: If an encrypted password file is set up, use as the first parameter in the command line to read the password from the full file path and name specified in <code>passwordFile</code> . See Suppressing Password Prompts in Planning Utilities .

Parameter	Description
<code>/U: <i>username</i></code>	The name of the Planning administrator
<code>/A: <i>sourceApplication</i></code>	The name of the source Planning application from which to push data
<code>/M: <i>applicationMapping</i></code>	The name of the application mapping that defines the push data range (see Mapping an Application for Reporting)
<code>[/C]</code>	Optional: Clears the data range in the target reporting application before pushing data

- If you are prompted, enter the password.

Note:

You can check the Job Console for the execution status. You can also view the results in the `PushData` log in the `Planning logs` directory. For the full path, see [About EPM Oracle Instance](#).

Examples:

```
PushData /U:admin /A:plnapp /M:LineItemsToExpenses /C
```

```
PushData -f: password.txt /U:admin /A:plnapp /M:LineItemsToExpenses /C
```

```
PushData /U:admin /A:plnapp /M:LineItemsToExpenses
```

Unsupported Features with Mapping Applications

Mapping an application to a reporting application does not support:

- Aggregation storage database outlines with the "Duplicate Members Allowed" option selected
- User variables
- Attribute dimensions
- Attribute member selections

Mapping Applications and Substitution Variables

When mapping an application that contains substitution variables, note:

- Substitution variables are checked when you click Push Data, not while application mappings are defined.
- For dimension-to-dimension mappings on the **Map Dimensions** tab:
 - The member selector displays Oracle Essbase substitution variables defined only for the Oracle Hyperion Planning application and for all Essbase applications.
 - For the Planning application, the variable name that is selected or typed is passed and evaluated when you click **Push Data**.
 - For the reporting application, the variable name is evaluated against the Planning application, and then the value is used in the Clear Data operation.
- For Point of View fields on the Point of View tab:

- For the Planning application, the member selector displays the Essbase substitution variables defined only for the Planning application and for all Essbase applications. The variable names that are selected or typed are passed and evaluated by Essbase when you click Push Data.
- For the reporting application, the member selector displays the substitution variables defined only for the reporting application and for all Essbase applications. It is evaluated against the reporting application, and the evaluated value is validated against the respective reporting dimension for the single member and no member function.
- If metadata in a reporting application has been modified, click **Refresh** before editing or pushing data to synchronize reporting dimensions and members with Planning. For example, if a dimension or member was added to a reporting application, clicking **Refresh** makes the member visible in Planning. Refreshing changes from reporting application metadata may cause mappings to become invalid.

Adding an Essbase Server for Reporting Applications

To add an Oracle Essbase server on which reporting application databases reside, on the **Add Essbase Server** dialog box:

1. In **Essbase Server**, type the server name.

To configure a data source to support Essbase failover in active-passive clustering mode, replace the **Essbase Server name** value with the APS URL followed by the Essbase cluster name; for example, if the APS URL is `http://hostname:13090/aps` and the Essbase cluster name is `EssbaseCluster-1`, then the value in the **Essbase Server name** field would be:

```
http://hostname:13090/aps/Essbase?clusterName=EssbaseCluster-1
```

Entering the Essbase cluster name without the APS URL in the **Essbase Server name** field is not supported in this release.

2. In **User Name**, type your user name.
3. In **Password**, type your password.
Your name and password are stored so you do not retype them in future sessions.
4. **Optional**: To test the connection to the Essbase server, click **Validate Connection**.
5. Click **OK**.

Editing an Essbase Server for Reporting Applications

You can use the **Edit Essbase Server** dialog box to change logon credentials and connectivity details such as server name and port numbers. For the default Oracle Essbase server, you can make updates in the **Manage Data Source** dialog box for the application's associated data source.

Note:

If you use the EPM System Configurator to change the Essbase server, you must update the server name within Oracle Hyperion Planning.

 **Tip:**

Using the Planning Upgrade Wizard, you can update Essbase server information for multiple reporting applications (see [Updating References to Reporting Essbase Servers](#)).

To edit an Essbase server displayed for reporting application databases, on the **Edit Essbase Server** dialog box:

1. In **Essbase Server**, select the server from the list of available servers.

To configure a data source to support Essbase failover in active-passive clustering mode, replace the **Essbase Server name** value with the APS URL followed by the Essbase cluster name; for example, if the APS URL is `http://hostname:13090/aps` and the Essbase cluster name is `EssbaseCluster-1`, then the value in the **Essbase Server name** field would be:

```
http://hostname:13090/aps/Essbase?clusterName=EssbaseCluster-1
```

Entering the Essbase cluster name without the APS URL in the **Essbase Server name** field is not supported in this release.

2. In **Server Name**, type the server name.
3. In **User Name**, type your user name.
4. In **Password**, type your password.
5. **Optional:** To test the connection to the Essbase server, click **Validate Connection**.
6. Click **OK**.

Removing an Essbase Server for Reporting Applications

To remove an Oracle Essbase server displayed for reporting application databases, on the **Delete Essbase Server** dialog box:

1. Select the server.
You cannot remove the default server, on which the current Oracle Hyperion Planning application resides.
2. Click **Delete**.
3. At the prompt, if you are sure you want to delete, click **OK**.

Deleting Application Information Using SQL

Oracle Hyperion Planning provides SQL files to delete this information:

- Account annotations. See [Deleting Account Annotations](#).
- Supporting detail associated with scenarios. See [Deleting Supporting Detail Associated With a Scenario](#).

You can use the Clear Cell Detail feature to clear account annotations, supporting detail, cell text, and cell-level documents. See [Clearing Cell Details](#).

Deleting Account Annotations

Use the `aadelete.sql` file, installed in the `sql` directory, to delete account annotations. It includes SQL queries that delete annotations for selected account names. To use the Clear Cell Detail feature to clear account annotations, see [Clearing Cell Details](#).

To delete account annotations associated with account names:

1. Stop the Web application server.
2. Update the SQL queries section of the `aadelete.sql` file corresponding to your type of relational database by substituting the name of the account whose annotations to delete.
3. Run queries in the `aadelete.sql` file appropriate for your relational database.

Example: deleting account annotations for `Account1`:

```
DELETE
FROM HSP_ACCOUNT_DESC
WHERE ACCOUNT_ID=(SELECT OBJECT_ID FROM HSP_OBJECT
WHERE OBJECT_NAME='ACCOUNT1')
INSERT INTO HSP_ACTION
(FROM_ID, TO_ID, ACTION_ID, OBJECT_TYPE, MESSAGE, ACTION_TIME, PRIMARY_KEY)
VALUES (0,0,2,18,NULL,GETDATE(),NULL)
```

Deleting Supporting Detail Associated With a Scenario

You can use the `sdddelete.sql` file, installed in the `sql` directory, to delete supporting detail associated with scenarios. It includes SQL queries that delete supporting detail for selected scenarios. To use the Clear Cell Detail feature to clear supporting detail, see [Clearing Cell Details](#).

To delete supporting detail associated with scenarios:

1. Stop the Web application server.
2. Update the SQL queries section of the `sdddelete.sql` file corresponding to your type of relational database by substituting the scenario name with the supporting detail to delete.
3. Run queries in the `sdddelete.sql` file appropriate for your relational database.
4. Start the Web application server.

Example: Deleting supporting detail associated with a scenario

Supporting detail for the Actual scenario is deleted:

```
DELETE
FROM HSP_COLUMN_DETAIL_ITEM
WHERE DETAIL_ID IN
(SELECT DETAIL_ID
FROM HSP_COLUMN_DETAIL
WHERE DIM1 =
```

```
(SELECT OBJECT_ID
FROM HSP_OBJECT
WHERE OBJECT_NAME = 'ACTUAL'));

DELETE
FROM HSP_COLUMN_DETAIL
WHERE DIM1 =
(SELECT OBJECT_ID
FROM HSP_OBJECT
WHERE object_name = 'Actual');
```

11

Working with Menus

Creating and Updating Menus

Administrators can create right-click menus and associate them with forms, enabling users to click rows or columns in forms and select menu items to:

- Launch another application, URL, or business rule, with or without runtime prompts
- Move to another form
- Move to Manage Approvals with a predefined scenario and version
- Open Job Console or Copy Version

The context of the right-click is relayed to the next action: the POV and the Page, the member the user clicked on, the members to the left (for rows), or above (for columns).

When designing forms, use Other Options to select menus available for Form menu item types. As you update an application, update the appropriate menus. For example, if you delete a business rule referenced by a menu, remove it from the menu.

To create, edit, or delete menus:

1. Select **Administration**, then **Manage**, then **Menus**.
2. Perform one action:
 - To create a menu, click **Create**, enter the menu's name, and then click **OK**.
 - To change a menu, select it, and then click **Edit**.
 - To delete menus, select them, click **Delete**, and then click **OK**.

Working with Menu Items

Edit Menu displays menu items on the current menu, including names, labels, required dimensions, icon, and type, such as URL, Form, Business Rule, Manage Approvals, Menu Header, Form, Job Console, and Copy Version.

To work with menu items:

1. Select **Administration**, then **Manage**, and then **Menus**.
2. Select a menu and click **Edit**.
3. **First time only:** To add the first item to the menu, click **Add Child** and **Save**.
4. Select a menu item and:
 - To add menu items below the selected item, click **Add Child** (available for Menu Header menu types).
 - To add menu items at the same level as the selected item, click **Add Sibling**.
 - To edit menu items, click **Edit**.

- To delete menu items, click **Delete**.
- To change the order of menu items within the same level, click the Up or Down Arrow. You can move multiple items.

Use **Edit Menu Item** to define the menu item properties.

5. Click **Save**.

Click **Save As** to save the current selections under a new menu name.

Adding or Changing Menu Items

To define menu items:

1. Select **Administration**, then **Manage**, and then **Menus**.
2. Select a menu and then click **Edit**.
3. Select the menu item and then click **Edit** or **Add Sibling**.
4. Define the menu item:

Table 11-1 Menu Items

Item	Description
Menu Item	Enter a unique name containing only alphanumeric and underscore characters, with no special characters or spaces
Label	Enter text to be displayed when the menu is selected. Spaces and special characters are allowed. Menu labels display in the user interface. Labels can be text or can reference a resource variable by name. For example, to set a menu's label to File, set it to <code>File</code> directly or to the name of a resource, such as <code>LABEL_FILE</code> , which can be localized.
Icon	Optional: In context of the Oracle Hyperion Planning server, enter the path and filename to a graphic to display by the menu. (In other words, the graphic or image referenced must be within the Planning Web application root folder.) For example: <code>Images/green.gif</code>

Table 11-1 (Cont.) Menu Items

Item	Description
Type	<p>Select the menu item type to determine available Properties. No properties are available for Menu Header.</p> <ul style="list-style-type: none"> • Form: Launch a selected form. The member selection context for the member, page, and POV is retained when users right-click in the source form. If the target form contains these dimension members on the page, its page is set to match the context. • URL: Navigate to the specified URL • Business Rule: Launch the selected business rule • Manage Approvals: Move to Manage Approvals to work with planning units • Menu Header: Create a menu under which you can create children menu items. To display a separator bar on the menu at this item, enter one hyphen as the Label. In this case, the Required Dimension list is not available. • Previous Form: Create a menu that returns the user to the previous form. • Job Console: Create a menu that opens the Job Console to view jobs for the current user for the specified Job Type and Status. • Copy Version: Create a menu that opens Copy Version to enable the user to copy data for the current form.
Required Parameter	<p>Select a dimension, or select an option for where the menu item displays: Point of View, Page, Row, Column, Members Only, Cell Only. For example, if you select Account, users can right-click Account members on a form to open the menu. If you select Row, the menu is available when users right-click a row. Selecting None makes the menu available whenever the user right-clicks in the form.</p>

5. Define menu item properties, which differ for menu item types:

Type	Options
Form	<ul style="list-style-type: none"> a. In Form Folder, select the folder containing the destination form. b. In Form, select the form.
URL	<ul style="list-style-type: none"> a. In URL, enter the complete URL to which to direct the user. For example: <code>http://server name/HFM/Logon/HsvLogon.asp</code>. b. Select Use Single Sign-on to append the SSO token to the URL. c. Select Include Context in URL to include the context.

Type	Options
Business Rule	<ul style="list-style-type: none"> a. In Plan Type, select the plan type for which the business rule is available. b. In Business Rules, select the business rule to launch. c. In View Type, select how to display runtime prompt pages: <ul style="list-style-type: none"> • Classic View: Use the default Planning view • Streamline View: Display each runtime prompt on a different line d. Optional: In Window Title, enter a title to display instead of Runtime Prompts. e. Optional: In OK Button Label, enter the text to display for the OK button. f. Optional: In Cancel Button Label, enter the text to display for the Cancel button. g. Optional: In Launch Confirmation Message, enter text to display when the business rule is invoked, but before it is launched. This option enables administrators to provide meaningful messages to planners about the consequences of launching business rules.
Manage Approvals	Specify the planning unit to which the user is directed by selecting a scenario and a version.
Previous Form	Enter the name of the menu item that will return the user to the previous form.
Copy Version	<p>Enable end users to use Copy Version to copy form data for the current form, including supporting details, annotations, cell text, and cell documents to another version. Select the following default values:</p> <ul style="list-style-type: none"> a. In Scenario, select the scenario from which to copy. b. In Copy From, select the version that contains the data to copy. c. In Copy To, select the version to which to copy the data.
Job Console	<p>Enable end users to view the Job Console for the specified job type and job. Select the following default values:</p> <ul style="list-style-type: none"> • Job Type • Status

6. Click **Save**.

12

Working with Planning Application Administration

About Creating Applications with Planning Application Administration

You can create applications using Oracle Hyperion Planning application administration. To use Planning application administration, you must be assigned the appropriate roles, as described in the *Oracle Enterprise Performance Management System User Security Administration Guide*.

Setting Up Applications Using Planning Application Administration

If you are using Oracle Hyperion Planning application administration, use the Planning Application Wizard to create and delete applications and to register with Oracle Hyperion Shared Services.

To set up applications using Planning application administration:

1. Perform a task:
 - From Planning, select **Administration**, then **Application**, and then **Manage Applications**.
 - From Oracle Hyperion Enterprise Performance Management Workspace, select **Navigate**, then **Administer**, and then **Planning Administration**.
2. See these topics to complete the task:
 - [Managing Applications](#)
 - [Managing Data Sources](#)
 - [Managing Upgrades](#)

Managing Applications

To use Oracle Hyperion Planning application administration to create, delete, and register Planning applications, see:

- [Creating Applications](#)
- [Deleting Applications](#)
- [Registering Applications](#)

Creating Applications

To create and update applications using Oracle Hyperion Planning application administration:

1. Start the Planning Application Wizard (see [Setting Up Applications Using Planning Application Administration](#)).
2. In **Manage Applications**, click **Actions**, and then select **Create**.
3. Define the application by completing information on the tabs. See:
 - [Selecting Application Information](#)
 - [Setting up the Calendar](#)
 - [Setting up Currencies](#)
 - [Specifying Plan Types](#)
 - [Reviewing Application Information](#)

Selecting Application Information

Specify the application name and description, register the application with Oracle Hyperion Shared Services and select the data source.

A default instance (cluster) is set up when you install and configure Oracle Hyperion Planning. To update clusters with the EPM System Configurator, see the *Oracle Enterprise Performance Management System Installation and Configuration Guide*.

To select application information:

1. In the Planning Application Wizard, select **Manage Applications**, and then under **Actions**, select **Create**.
See [Setting Up Applications Using Planning Application Administration](#).
2. Click **Select**.
3. Select a data source.
See [Managing Data Sources](#).
4. Enter the name of the application.
The name can contain up to eight characters. It must not be the same as an existing Oracle Essbase application.
5. Enter a description of the application.
6. For **Application Type**, select which type of application you are creating. For Planning applications, select **General**.
7. Select a Shared Services application group.
8. Click **Next**.

Setting up the Calendar

The calendar establishes the application's base time periods, first fiscal year and month, and total number of years. Select the base time period and monthly distribution pattern based on the number of fiscal weeks in a month. The base time period options are the bottom-level time periods in the application. You can create a custom base time period, such as weeks or days. Use monthly distribution patterns to determine how data entered into a summary time period is distributed or spread among the base time period that you select. Users can enter data into summary time periods, such as years or quarters. Oracle Hyperion Planning distributes these values over the base time periods that constitute the summary time period.

For the fiscal year, you can set the fiscal year first month and specify whether the fiscal year starts from the same calendar year or the previous calendar year. You can later set up calculations based on the calendar year, for example, using formula expressions. When setting up formulas for Planning applications, keep in mind that formula expressions such as [TPDate] and [FirstDate] produce different results if the application starts in the same calendar year or previous calendar year. See [Working with Formula Expressions](#).

The following table provides examples of how the Fiscal Year First Month and the Fiscal Start Year options affect the calendar for the application, assuming the Fiscal Start Year is 2012.

Table 12-1 Examples of Fiscal Year First Month and Fiscal Start Year Calendar Options

Fiscal Year First Month	Fiscal Start Year	Period - Year	Years Dimension
January	Same Calendar Year	Jan-12 to Dec-12	FY12
July	Same Calendar Year	Jul-12 to Jun-13	FY12
July	Previous Calendar Year	Jul-11 to Jun-12	FY12
February	Same Calendar Year	Feb-12 to Jan-13	FY12
February	Previous Calendar Year	Feb-11 to Jan-12	FY12
December	Same Calendar Year	Dec-12 to Nov-13	FY12
December	Previous Calendar Year	Dec-11 to Nov-12	FY12

To set up the calendar:

- In the Planning Application Wizard, click **Calendar**.
 See [Setting Up Applications Using Planning Application Administration](#).
- To set how calendars roll up, select a **Base Time Period** option:
 - 12 Months:** Four quarters per year; months roll up into parent quarters and quarters into years.
 - Quarters:** Quarters roll up into years.
 - Custom:** A custom time period, such as weeks or days.
- Select the **First Fiscal Year**.
 This selection defines the starting fiscal year for the application. You cannot change it after creating the application. Before specifying the first fiscal year, consider how much historical data your organization needs and wants in the application.
- Select the **Number of Fiscal Years** for the application.
 This selection defines the number of years in the calendar. You can add more years to the calendar after the application is created.
- Select the **Fiscal Year First Month**.
 This selection is the first month in the fiscal year for the application, which depends on the First Fiscal Year option.
- Select the **Fiscal Year Start Date**:
 - Same Calendar Year.** Set the fiscal year to start with the current calendar year. For example, selecting the year 2014 with a starting period of Jun creates the starting year as FY14, defined as Jun-14 to May-15.
 - Previous Calendar Year.** Set the year to start with the previous calendar year. For example, selecting the year 2013 with a starting period of Jun creates the starting year as FY13, defined as Jun-12 to May-13.

For an application starting in January, only the **Same Calendar Year** option is available.

7. If you set the base time period as **12 Months**, select a **Weekly Distribution** option: **Even**, **445**, **454**, or **544**.

Weekly distribution sets the monthly distribution pattern, based on the number of fiscal weeks in a month. This selection determines how data in summary time periods spreads within the base time period. When users enter data into summary time periods, such as quarters, the value is distributed over base time periods in the summary time period.

If you select a weekly distribution pattern other than **Even**, Planning treats quarterly values as if they were divided into 13 weeks and distributes weeks according to the selected pattern. For example, if you select **5-4-4**, the first month in a quarter has five weeks, and the last two months in the quarter have four weeks.

8. To create an All Years parent member that includes all years, select **Yes** for **All Years parent**.

The All Years parent member enables users to view the accumulated data across multiple years, for example, a project's total cost up to its end date. The parent member does not include the No Year member, if one is defined for the application.

9. Click **Next**.

Setting up Currencies

Specify the default currency for entities in the application, and establish if the application supports currency conversions. Multiple currency support (also called currency overrides) is available for level 0 members, regardless of their base currency.

To set up currencies:

1. In the Oracle Hyperion Planning Application Wizard, click **Currencies**.
See [Setting Up Applications Using Planning Application Administration](#).
2. Select the default currency for entities in the application.
3. Specify if the application supports multiple currencies by selecting **Yes** for multicurrency applications or **No** for single-currency applications.

After the application is created, you cannot change this option. Multiple currency support is available for level 0 members, regardless of base currency. If you select **Yes**, two additional dimensions are created, Currency and HSP_Rates.

4. Click **Next**.

Specifying Plan Types

Specify one to three plan types for the application. A separate Oracle Essbase database is created for each plan type. You cannot delete a plan type or change the plan type name after creating an application.

As you create accounts, entities, and other elements of the application, you associate them with plan types, so the database for each plan type contains only information relevant to the plan type. This optimizes application design, size, and performance.

 **Note:**

You can specify up to three generic plan types for an application. If you select one block storage plan type during application creation, then you are allowed to add two more from the plan type editor for a total of three.

Oracle Hyperion Planning allows up to three generic plan types total and one aggregate storage database for each block storage database plus one additional aggregate storage database for consolidation purposes.

To select plan types:

1. In the Planning Application Wizard, click **Plan Types**.
See [Setting Up Applications Using Planning Application Administration](#).
2. For each plan type in the application, select the **Plan Type** and specify a plan name.
You must select least one Planning plan type. You can have up to three Planning plan types, and names can contain up to eight characters. (Although it is possible to enter more than eight bytes using single-byte and double-byte characters, an error message displays when the Essbase database is created.)
3. If you are creating an aggregate storage plan type, select ASO1, and then specify an application name.

 **Note:**

Essbase requires a separate application for each aggregate storage database.

4. Click **Next** to review application information and create the application.

Reviewing Application Information

To review application information before creating or updating the application:

1. In the Oracle Hyperion Planning Application Wizard, click **Finish**.
See [Setting Up Applications Using Planning Application Administration](#).
2. Review the selected settings. Some settings cannot be changed after the application is created.
3. **Optional:** To modify application settings, click **Previous**, update the application information, and click **Finish** to check the updated settings.
4. To create the application, click **Create**.

Deleting Applications

You can use the Oracle Hyperion Planning Application Wizard to delete applications that use Planning application administration.

 **Caution:**

Oracle recommends backing up the application before deleting. See [Backing Up Applications and Application Databases](#).

To delete applications:

1. Back up the application.
2. Start the Planning Application Wizard (see [Setting Up Applications Using Planning Application Administration](#)).
3. In **Manage Applications**, select an application, click **Actions**, and then select **Delete**.
4. At the confirmation message, click **OK** to proceed with the deletion.

Registering Applications

You can use the Oracle Hyperion Planning Application Wizard to reregister the Oracle Hyperion Shared Services application group for an application.

To register applications:

1. Start the Planning Application Wizard (see [Setting Up Applications Using Planning Application Administration](#)).
2. In **Manage Applications**, select an application, then click **Actions**, and then select **Register**.
3. In **Register**, select the **Shared Services Project**, and then click **OK**.

Managing Data Sources

Each Oracle Hyperion Planning application must be associated with its own data source, which links the relational database and the Oracle Essbase server. To use the Planning Application Wizard to create, update, test, and delete data sources, see:

- [Creating Data Sources](#)
- [Editing Data Sources](#)
- [Checking Connections](#)
- [Deleting Data Sources](#)

Creating Data Sources

Each Oracle Hyperion Planning application requires its own data source that points to a separate relational database. If you create a new data source and associate it with an existing relational schema that is used by another application, the application creation process will overwrite the relational tables for the previously created application.

To create data sources for application administration, enter the data source name and description, select a relational database, and specify details for the relational database and Oracle Essbase server. The relational database and Essbase server password information is stored as encrypted. You can also set the application to Unicode mode. Unicode-mode applications support multiple character sets. When Essbase works with Unicode-mode applications, it uses the UTF-8 encoding form to interpret and store character text. Character-

based artifacts in Unicode-mode applications, such as member and alias names, can include characters from different languages. For more information, see the *Oracle Essbase Database Administrator's Guide*.

To create data sources for application administration:

1. In the Planning Application Wizard, click **Manage Data Source**, and then under **Actions**, select **Create**. See [Setting Up Applications Using Planning Application Administration](#). Within Planning, you can also select **Administration**, then **Manage Data Source**, then click **Actions**, and then select **Create**.
2. Enter the data source name and description.
3. Specify these application database details:
 - **Database**—Relational database
 - **Server**—Server hosting the database
 - **Port**—Port (for default ports, see *Oracle Hyperion Enterprise Performance Management System Installation Start Here*)
 - **Service Name or SID**—Oracle service name or SID (for example, orcl)
 - **User**—Database username
 - **Password**—Database password
4. **For advanced users only:** Optionally, create the data source using a custom URL instead of the information in the **Server** and **Port** fields. Click **Custom**, and then enter the URL for the data source in **Connection URL**.

For detailed information on configuring the database with the EPM System Configurator and examples of URLs, see the *Oracle Enterprise Performance Management System Installation and Configuration Guide*.

Entering a custom URL overrides previous connection settings for server and port. To remove the custom URL and return to the previous server and port settings, clear the **Custom** check box. If Oracle RAC configuration is being used, the RAC details must be provided in **Custom URL** during Planning data source creation.

5. Click **Validate**, and fix any issues that are noted.
6. Specify these Essbase server details:
 - **Server**—Server name (If the default port number Essbase Agent port was changed to another number during configuration, the Oracle Essbase server must have the format `ServerName:newPortNumber`. For example, if the port number was set to 1400 during configuration and the server is run locally, the server name uses this format, `ServerName:1400.`)
 - **User**—Server username
 - **Password**—Server password
 - **Optional**—To set the application to Unicode mode, select **Unicode**.

For more information, see the *Oracle Essbase Database Administrator's Guide*.
7. Click **Validate**, and fix any issues that are noted.
8. Click **Save**.

Editing Data Sources

For Oracle Hyperion Planning application administration, you can use the wizard to update the data source name, description, relational database, and details for the relational database and Oracle Essbase server.

To edit data sources:

1. In the Planning Application Wizard, click **Manage Data Source**. See [Setting Up Applications Using Planning Application Administration](#). Within Planning, you can also select **Administration**, and then select **Manage Data Source**.
2. Select a data source, click **Actions**, and then select **Edit**.
3. Update the data source name and description.
4. Specify the application database details:
 - **Server**: Server hosting the database
 - **Port**: Port. For information about default ports and how to change them, see *Oracle Hyperion Enterprise Performance Management System Installation Start Here*.
 - **Database**: Database name
 - **User**: Database username
 - **Password**: Database password
5. **For advanced users only**: Optionally, create the data source using a custom URL instead of the information in the **Server** and **Port** fields. Click **Custom**, and then enter the URL for the data source in **Connection URL**.

For detailed information on configuring the database with the EPM System Configurator and examples of URLs, see the *Oracle Enterprise Performance Management System Installation and Configuration Guide*.

Entering a custom URL overrides previous connection settings for server and port. To remove the custom URL and return to the previous server and port settings, clear the **Custom** check box.

6. Click **Validate**, and fix any issues that are noted.
7. Specify Essbase server details:
 - **Server**: Server name
 - **User**: Server username
 - **Password**: Server password
8. **Optional**: To set the application to Unicode mode, select **Unicode Mode**.
For more information, see the *Oracle Essbase Database Administrator's Guide*.
9. Click **Validate**, and fix any issues that are noted.
10. Click **Save**.
11. At the message that the data source was updated successfully, click the X to close the message.

Checking Connections

You can test the connections to the database or Oracle Essbase.

To check connections:

1. In the Oracle Hyperion Planning Application Wizard, click **Manage Data Source**.
See [Setting Up Applications Using Planning Application Administration](#).
2. Specify the application database details:
 - **Server:** Server hosting the database
 - **Port:** Port. For information about default ports and how to change them, see *Oracle Hyperion Enterprise Performance Management System Installation Start Here*.
 - **Database:** Database name
 - **User:** Database username
 - **Password:** Database password
3. **For advanced users only:** Optionally, create the data source using a custom URL instead of the information in the **Server** and **Port** fields. Click **Custom**, and then enter the URL for the data source in **Connection URL**.

For detailed information on configuring the database with the EPM System Configurator and examples of URLs, see the *Oracle Enterprise Performance Management System Installation and Configuration Guide*.

Entering a custom URL overrides previous connection settings for server and port. To remove the custom URL and return to the previous server and port settings, clear the **Custom** check box.

4. Click **Validate**, and fix any issues that are noted.
5. Specify Essbase server details:
 - **Server:** Server name
 - **User:** Server username
 - **Password:** Server password
6. **Optional:** To set the application to Unicode mode, select **Unicode Mode**.
For more information, see the *Oracle Essbase Database Administrator's Guide*.
7. Click **Validate**, and fix any issues that are noted.
8. Click **Save**.
9. At the message that the data source was updated successfully, click the X to close the message.

Deleting Data Sources

For Oracle Hyperion Planning application administration, you can use the Planning Application Wizard to delete data sources that are not associated with an application.

To delete data sources:

1. In the Planning Application Wizard, click **Manage Data Source**. See [Setting Up Applications Using Planning Application Administration](#). Within Planning, you can also select **Administration**, and then select **Manage Data Source**.
2. Select the data source to delete, click **Actions**, and then select **Delete**.
3. At the message that the data source was deleted successfully, click the X to close the message.

Managing Upgrades

Use the Oracle Hyperion Planning Upgrade Wizard to update references to rehosted data sources and to upgrade Planning applications. For example, if the Oracle Essbase Server host and port have changed since the earlier release, you must update data sources and update mappings from Planning applications to reporting applications. If the relational data source has changed, you must update data source connections. The Planning Upgrade Wizard enables you to update multiple data sources and reporting applications simultaneously.

The Upgrade Wizard also upgrades applications using Planning application administration.

- For updating references to rehosted relational databases and Essbase servers, see [Updating References to Data Sources](#).
- For updating references to rehosted Essbase servers that have reporting applications, see [Updating References to Reporting Essbase Servers](#).
- For upgrading Planning applications created in Planning application administration, see [Upgrading Applications](#).

Updating References to Data Sources

Use the Oracle Hyperion Planning Upgrade Wizard to update references to rehosted relational databases and Oracle Essbase servers.

 **Note:**

Before updating references to data sources, ensure that the Essbase server and the relational database are running.

To update references to data sources:

1. In the Planning Application Wizard, click **Upgrade Wizard**.
2. On the **Update Data Sources** tab, review the relational database information and the Essbase server information for each Planning data source. If the Essbase server host and port have changed during upgrade, or if the relational database has moved to a new host, update the information.
 - To update multiple relational databases with the same information:
 - a. On the bottom of the page, select the data sources to which to apply relational database updates.
 - b. Under **Update Relational Information**, enter the new database information.
 - c. Click **Apply to Selected**.
 - To update multiple Essbase servers with the same information:
 - a. On the bottom of the page, select the data sources to which to apply Essbase information updates.
 - b. Under **Update Essbase Information**, enter the new Essbase server information.

When using Essbase failover in active-passive clustering mode, note that you can specify the APS URL followed by the Essbase cluster name; for example, if the

APS URL is `http://<hostname>:13090/aps` and the Essbase cluster name is `EssbaseCluster-1`, then the value in the Essbase Server name field would be:

`http://<hostname>:13090/aps/Essbase?clusterName=EssbaseCluster-1`

Entering the Essbase cluster name without the APS URL is not supported in this release.

- c. Click **Apply to Selected**.
 - To update each data source individually:
 - a. On the bottom of the page, select the data sources to which to apply the updates.
 - b. Enter the new information for each data source.
3. Choose an option under **Actions**
 - Click **Reset** to undo the updates.
 - Click **Validate** to test the connections to the selected data sources, and fix any issues that are noted.
 - Click **Save** to save the updates to the selected data sources.
4. Click **Next** to proceed to the **Upgrade Applications** tab (see [Upgrading Applications](#)).



Note:

If you do not see the updates reflected in the Planning application, stop and then restart the Planning server.

Upgrading Applications

Use the Oracle Hyperion Planning Upgrade Wizard to upgrade Planning applications created using Planning application administration. Instructions for this procedure are available in the *Oracle Enterprise Performance Management System Installation and Configuration Guide*.



Note:

You must update references to rehosted data sources before you upgrade applications. See [Updating References to Data Sources](#) and [Updating References to Reporting Essbase Servers](#).

Oracle Hyperion Public Sector Planning and Budgeting users must perform additional data migration tasks after an upgrade. Instructions for migrating existing application metadata and artifacts are provided in Appendix B of the *Oracle Hyperion Public Sector Planning and Budgeting User's Guide*.

Updating References to Reporting Essbase Servers

Use the Oracle Hyperion Planning Upgrade Wizard to update references to Oracle Essbase servers that are used in reporting applications.

 **Note:**

Before updating references to reporting Essbase servers, ensure that the Essbase server and the relational database are running.

To update references to Essbase servers that have reporting applications:

1. In the Planning Application Wizard, click **Upgrade Wizard**.
2. On the **Update Reporting Essbase Servers** tab, review or update the Essbase server information for reporting applications.

Only applications that have already been upgraded and that have reporting applications created on an Essbase server (other than the default Essbase server) are listed.

- To update multiple reporting applications with the same information:
 - a. On the bottom of the page, select the applications to which to apply the updates.
 - b. Under **Update Reporting Essbase Information**, enter the new Essbase server information.
 - c. Click **Apply to Selected**.
 - To update each reporting application individually:
 - a. On the bottom of the page, select the reporting applications to which to apply the updates.
 - b. Enter the new Essbase server information for each application.
3. Choose an option under **Actions**:
 - Click **Reset** to undo the updates.
 - Click **Validate** to test the connections to the selected Essbase servers, and fix any issues that are noted.
 - Click **Save** to save the updates to the selected Essbase servers.
 4. Click **Back** to switch to the **Upgrade Applications** tab (see [Upgrading Applications](#)).

 **Note:**

If you do not see the updates reflected in the Planning application, stop and then restart the Planning server.

About Monitoring and Optimizing Performance Using the Application Monitor

The application monitor enables administrators, at design-time, to identify and resolve design flaws before an application is placed in production (and on an ongoing basis, as the application evolves with new members and data is added to the application) and used by end users. Administrators can use the application monitor to evaluate the following:

- An entire application

- Types of artifacts such as forms and planning units
- Individual "cherry picked" artifacts

The application monitor identifies changes that Administrators can make at design-time to resolve artifacts with design flaws, and provides easy access to the editors required to modify them. At runtime, internal processes perform checks that may deny system request if artifacts need to be modified. The application monitor evaluates your application and artifacts based on the user an administrator selects. This enables you to determine the potential problems that may occur at runtime given the individual user's variables and security access as part of the analysis.

Assumptions

Oracle can only provide general guidelines about how to design your application and artifacts to ensure reasonable performance. Oracle cannot specify precise value ranges for artifacts and operations.

How the Application Monitor Works

The application monitor evaluates applications as follows:

- At design-time—Analyzes the application or specific artifacts in terms of all possible problems that could occur. To determine this worst possible situation, suppression options (such as those used with forms) are ignored. The application monitor provides information about how to fix the design flaws it encounters.
- At runtime—Internal governors perform checks that may prevent users from performing particular actions if the design flaws found are not resolved.

Use the application monitor to analyze the following:

- Block storage plan types
- Dimensions
- Simple forms
- Standalone business rules
- Planning unit hierarchies
- Reporting mappings
- Export data functionality

To optimize the user experience and performance of the application and to ensure that the application monitor provides a warning when thresholds are in danger of exceeding optimal limits, you must set appropriate thresholds for the artifacts listed above. The threshold values you choose are dependent on many factors; for example, hardware, number of users, and so on, and these numbers will change based on your requirements. For example, if you want your application to be restricted to a five-year plan, set the Years threshold to five. Then the application monitor will warn you when this threshold is violated.

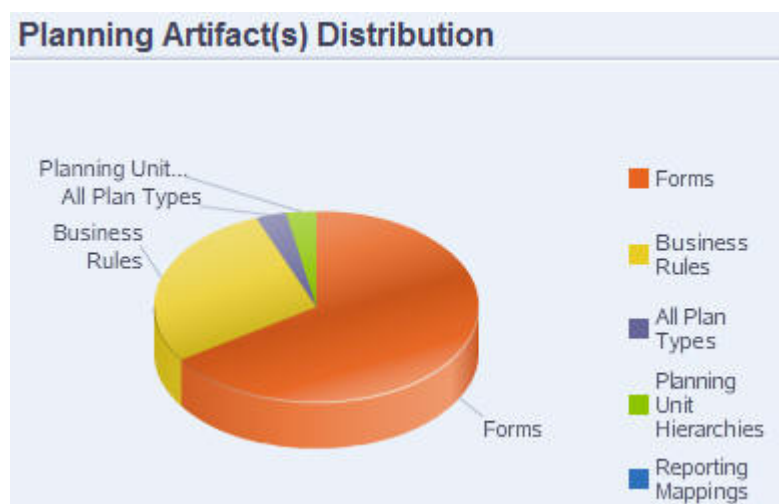
The system is highly sensitive to the artifact types monitored by the application monitor. These thresholds must be set based on the limits that were determined during user acceptance testing. See [Setting Limits for Application Monitor Warning and Error Thresholds](#).

Using the Application Monitor Graphs

Artifacts that may cause performance issues and that should be modified are yellow, and those that will cause performance issues and must be redesigned are red. Use the pie graphs as follows:

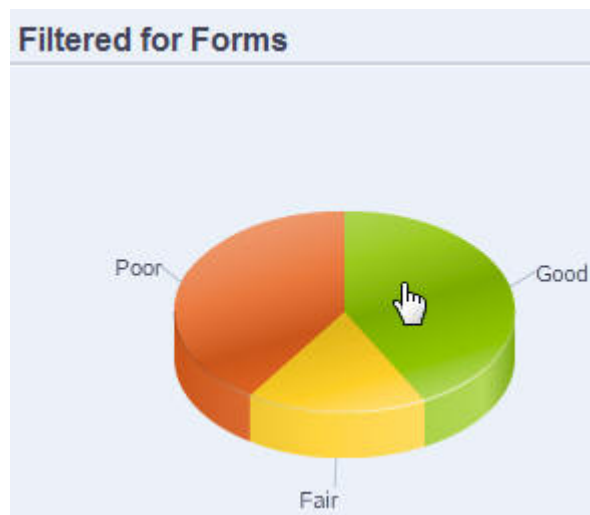
- **Planning Artifacts Distribution** —Depicts how much of your application is comprised of various artifact types. Place your mouse over portions of the graph to view how many artifacts of each type were analyzed by the application monitor. For example, if you mouse over forms and the pop-up value is 55, then 55 forms were analyzed. Click each portion of the graph to view the compliance status of each artifact type. This displays another pie graph that represents how many artifacts of each type are in acceptable, warning, or error status.

Artifact Distribution Graph:



- **Filtered For <Artifact>**—Click the graph portions that represent the artifact status, and to view and drill down on artifacts with a particular status. For example, to view information for only planning unit hierarchies in error status, click the red portion of the graph.

Filtered Graph:







Setting Limits for Application Monitor Warning and Error Thresholds

No thresholds are included with Oracle Hyperion Planning, however you can set warning and error thresholds for artifacts by adding application properties so they can be effectively analyzed by the application monitor. If the warning thresholds you set are exceeded, in application monitor, a warning will be issued and the artifacts will turn yellow, and then they can be modified. If the error thresholds you set are exceeded, the artifacts will turn red in application monitor and system requests will be denied at runtime.

To set application warning and error thresholds, see [Setting Threshold Limits for Applications](#).



Launching the Application Monitor

To launch the application monitor:

1. After logging in, select **Diagnostics** in the left hand pane.
2. In **Select User**, click  to select the user whose settings you want to use to perform the analysis, and then click .
3. Perform a task:
 - To analyze and troubleshoot all application artifacts, select **Application Monitor - All Artifacts**. To the right, a pie graph depicts the distribution of artifacts in the application. The lower grid represents the integrity of all application artifacts. Yellow and red indicate artifacts that you should or must modify for optimal performance.
 - To analyze specific types of artifacts, such as forms or planning unit hierarchies, select the type in **Diagnostics**. For example, to ensure that business rules are designed well and will not cause performance issues, click  in **Diagnostics**. If a large portion of the graph is yellow or red, you should modify your rules. More information about all artifacts of that type display in the grid below.
 - To "cherry pick" and analyze one particular artifact, find the artifact by specifying its name, start date, end date, and type in **Search**, select the artifact, and then click . Information about the artifact displays in the lower grid.

 **Tip:**

For information about what the pie graphs display and how to drill down to access more information, see [Using the Application Monitor Graphs](#).

4. Click  or select **Actions**, and then **Run Application Monitor**.
5. If the **View Details** column does not display in the grid, select **View**, then **Columns**, and then **View Details**. This also enables you to display or conceal different types of artifact data.
6. To display information about how to fix artifacts, click  in **View Details**. Then click artifacts to launch editors that enable you to modify them.

Modifying Artifacts for Optimal Performance

After reviewing the details provided by the application monitor that tell you how to modify your artifacts, edit them as follows:

- Simple forms—Clicking on the form artifact name launches the **Form Designer** in a new tab. Edit forms as described in [Managing Forms](#).Chapter 6, "Managing Forms."
- Business Rules—Clicking a business rule name launches the **System View** which displays all application artifacts that you can access. You can perform tasks such as:
 - Use the upper **Rule Designer** to view the individual components of the rule, such as conditions, commands, and scripts. Click each component to modify them in the lower portion of the page. You can also insert and remove components like formulas, system templates, and scripts by dragging and dropping them from the **Rule Pallet** and **Existing Object** panes to the left.
 - From **Designer**, select **Edit Script** to modify and format the script by performing tasks such as inserting functions, editing templates, inserting variables, inserting member ranges, use comments, and validate syntax.
 - Create, open, delete, and refresh artifacts, pages, and other documents.
 - Import and export business rules and other objects.
 - Determine how and where artifacts are used.

See *Oracle Hyperion Calculation Manager Designer's Guide* and [Understanding Runtime Prompts](#).
- Planning unit hierarchies—Clicking a planning unit hierarchy artifact name launches the **Planning United Hierarchy Designer** in a new tab. Edit planning units as described in [Creating Planning Unit Hierarchies](#).
- Plan types—Clicking the plan type artifact name launches the **Performance Settings** tab of the dimension editor in a new tab. Edit the dimensional outline or dimensional sparsity as described in [Working with Dimensions](#).
- Reporting Mappings—Clicking the reporting mapping artifact name launches the **Report Mapping Designer** in a new tab. Edit reporting mappings as described in [Mapping an Application for Reporting](#).

Working with Alias Tables

You can create and update alias tables, and set a default alias table for the application. Follow naming conventions in [Naming Restrictions](#).

About Alias Tables

You can assign alternate names, or aliases, to Oracle Hyperion Planning Account, Currency, Entity, Scenario, Period, Version, Years, and user-defined dimension members. Planning allows up to 10 aliases per dimension member, including the default alias.

When creating a Planning application, Oracle Essbase creates an empty default alias table in the database outline. If you do not create other alias tables, all aliases are stored in this default table. You cannot delete the default table.

You can create up to nine alias tables in Essbase. If you add or change aliases or alias tables, you must refresh the application. Changes are not in effect until the database is updated. To view the database outline, open Administration Services Console, select Outline, then Aliases, then Set Table, and select an alias table. You can use only alias tables created from within Planning. Alias tables created outside Planning are deleted during application refresh.

Multiple alias tables support, for example, these language combinations:

- English, French, German, Spanish, and Italian
- Japanese and English
- Korean and English
- Turkish and English



Note:

Alias table support is not restricted to these language combinations.

You can set alias tables to display members in an application. Planners can set alias tables in preferences.

Creating Alias Tables

To create alias tables:

1. Select **Administration**, then **Manage**, then **Alias Tables**.
2. Click **Add**.
3. In **Add - Alias Table**, enter a name.
4. Click **OK**.

Editing or Renaming Alias Tables

To edit or rename alias tables:

1. **Administration**, then **Manage**, then **Alias Tables**.
2. Select the alias table.
3. Click **Edit**.
4. For **Edit - Alias Table**, enter a name.
5. Click **OK**.

Deleting Alias Tables

To delete alias tables:

1. **Administration**, then **Manage**, then **Alias Tables**.
2. Select the alias table.
You cannot delete the default alias table.
3. Click **Delete**.
4. Click **OK**.

Clearing Alias Tables

You can clear the contents of alias tables.

To clear alias tables:

1. **Administration**, then **Manage**, then **Alias Tables**.
2. Select the alias table to clear.
Clearing the alias table removes the contents of the table but does not remove the table.
3. Click **Clear Values**.
4. Click **OK**.

Copying Alias Tables

To copy alias tables:

1. **Administration**, then **Manage**, then **Alias Tables**.
2. Select the alias table.
3. Click **Copy**.
4. Select the destination alias table.
The destination alias table must exist. Copying does not create tables.
5. Click **Copy**.

Specifying a Default Alias Table, and Setting Member and Alias Display Options

If you create alias tables with aliases for Account, Currency, Entity, Scenario, Period, Version, Years, and user-defined dimensions and members, you can select a default alias table for the

application. Users can set preferences for which set of aliases (stored in an alias table) to use for displaying member and dimension names.

To select the application's default alias table:

1. Select **Administration**, then **Application**, then **Settings**.
2. Select **Current Application Defaults**, and then select **Application Settings**.
3. In **Alias Table**, select a default alias table.
4. In **Member Name\Alias Display**, select the option that enables the kind of member data to be displayed on the Member Selector throughout your application:
 - **Default**—The data determined by the form, grid, or dimension settings
 - **Member name** —Just member names
 - **Alias**—Just member aliases, if defined
 - **Member name : Alias**—Names followed by aliases, if defined
 - **Alias:Member name**—Alias, if defined, followed by the names
5. Click **Save** or **Reset**.

Working with Dimensions

Dimension Overview

Dimensions categorize data values. Seven dimensions are included with Oracle Hyperion Planning: Account, Entity, Scenario, Version, Period, Years, and Currency. You can create up to 13 user-defined custom dimensions.

About Dimensions and Members

Members are components of dimensions.

About Sparse and Dense Dimensions

Sparse dimensions lack data values for the majority of member combinations. Dense dimensions have data values for the majority of member combinations. At least one dense dimension is required. Custom attributes cannot be assigned to dense dimensions. Oracle Hyperion Planning designates the Account and Period dimensions as dense, and the remaining dimensions as sparse. To optimize performance for sparse dimensions, Planning searches for and calculates only occupied data values in each dimension combination, reducing calculation time and disk usage. You can modify these settings. See [About Reordering Dimensions](#) and [Setting Dimension Density and Order](#).

About Dimension Hierarchies

Dimension hierarchies define structural and mathematical relationships, and consolidations between members in the database. Relationships are represented graphically in a collapsible hierarchy diagram. The levels below the database name are dimensions, and the levels below each dimension are members.

The Period dimension can contain the member YearTotal, which contains members Q1, Q2, Q3, and Q4. Members Q1, Q2, Q3, and Q4 contain their own members for the corresponding

months in the year. To consolidate data values in the Period dimension, roll up monthly data values to get quarterly data values, and quarterly data values to get yearly data values.

Members of the same level that belong to the same dimension or member are called siblings. For example, Q1, Q2, Q3, and Q4 are siblings because they are at the same level in the hierarchy, and are members of the same member, YearTotal.


The members of a dimension are called children of the dimension. Members that belong to a member are called children of that member. The member YearTotal is a child of Period, the members of Q1, Q2, Q3, and Q4 are children of YearTotal, and Jan, Feb, and Mar are children of Q1. Q1 is the parent of Jan, Feb, and Mar, YearTotal is the parent of Q1, Q2, Q3, and Q4, and Period is the parent of YearTotal.

Working with Dimension Hierarchies


Task	Topic
Specify or change dimension properties.	Click Edit .
Add a dimension.	Click Add Dimension .
Search for a dimension member.	See Finding Dimensions or Members .
Expand or collapse the dimension hierarchy.	Click Expand or Collapse .
Add or edit a dimension member.	Click Add Child or Add Sibling .
Move a dimension member.	See Moving Members Within the Dimension Hierarchy .
Delete a dimension member.	See Deleting Members .
Assign access to a dimension member.	See Assigning Access to Members and Business Rules .
View a member's ancestors.	Click Show Ancestors .

Expanding and Collapsing Dimension Hierarchies

To expand dimensions or members:

1. Select **Administration**, then **Manage**, then **Dimensions**.
2. For **Dimension**, select the dimension and member to expand.
3. Perform one action:
 - Click **Expand**.
 - Click .
 - Click the closed folder.

To collapse dimensions or members:

1. Select **Administration**, then **Manage**, then **Dimensions**.
2. For **Dimension**, select the dimension to collapse.
3. Perform one action:
 - Click **Collapse**.
 - Press the **Left Arrow**.
 - Click .
 - Click the open folders.

Navigating Dimension Hierarchies

- Press Up Arrow to move to the previous member.
- Press Down Arrow to move to the next member.
- In Page, enter the page to view and click Go or press Enter.
- Click Start, Prev, Next, or End to view other pages.

By default, 14 members are displayed per page. You can change this by setting preferences for Show the Specified Members on Each Dimensions Page.

Filtering the Dimension View by Plan Type

You can filter the dimension view by plan type. When you select a plan type, only members used in that plan type are displayed on the **Dimensions** page.



To filter the dimension view by plan type:

1. Select **Administration**, then **Manage**, then **Dimensions**.
2. For **Plan Types**, select the plan type.

Oracle Hyperion Planning displays only the members used in the selected plan type.

Finding Dimensions or Members

To find dimension members in dimension hierarchies:

1. Select **Administration**, then **Manage**, then **Dimensions**.
2. For **Dimension**, select the dimension for the member.
3. For **Search**, select name, alias, or both.
4. Enter the member name, alias, or partial string for which to search.
5. Click **Search Down**  or **Search Up** .



Sorting Members

You can sort members in ascending or descending order, by children or descendants. Sorting members affects the outline.

To sort members:

1. Select **Administration**, then **Manage**, then **Dimensions**.
2. For **Dimension**, select the dimension for the members.
3. On **Dimensions**, select the members whose children or descendants you want to sort.
4. For **Sort**, select children or descendants.

Sorting by children affects only members in the level immediately below the selected member. Sorting by descendants affects all descendants of the selected member.



5. Click  to sort by ascending order or  to sort by descending order.
6. Click **OK**.

The next time you create or refresh the database, the outline is generated with members in the order that is displayed.

Moving Members Within the Dimension Hierarchy

You can move one member or a group of members in the same branch. If you move Account members whose Valid For Plan Type settings differ from their new parent, the moved member settings change to match the setting of the new parents. If you move members whose Source Plan Type settings differ from their new parent, the moved member Source Plan Type is reset to match the first valid plan type.

To move members or branches among siblings:

1. Select **Administration**, then **Manage**, then **Dimensions**.
2. For **Dimension**, select the dimension for the members to move.
3. Select the member or branch to move.
4. Perform one action:
 - Click  to move the member up one position.
 - Click  to move the member down one position.

To move members, including parents and children:

1. Select **Administration**, then **Manage**, then **Dimensions**.
2. For **Dimension**, select the dimension with the members to move.
3. Select the member or branch to move.
4. Click **Cut**.

You cannot Cut members after adding or editing dimensions, navigating to different pages, deleting members, or logging off Oracle Hyperion Planning. Not available for root dimension members.
5. Click the destination level under which to move the members.
6. Click **Paste**.
7. Click **OK**.
8. Update and validate business rules and reports.

Viewing a Member's Ancestors

To view a member's ancestors:

1. Select **Administration**, then **Manage**, then **Dimensions**.
2. For **Dimension**, select a dimension.
3. Select the member in the dimension hierarchy.
4. Click **Show Ancestors**.
5. Click **Close**.

Determining Where Members Are Used in an Application

To view where members are used in an application:

1. Select **Administration**, then **Manage**, then **Dimensions**.
2. Select the dimension whose member's usage you want to view.
3. Click **Show Usage**.
4. At the bottom of the **Member Usage** window, select where in the application to view the member's usage.
Options are appropriate for the selected dimension member.
5. Click **Go**.
6. Click **Close**.

About Custom Dimensions

Oracle Hyperion Planning includes two custom dimensions: Account and Entity. You can edit the names of these dimensions, and create up to 13 user-defined dimensions. Use Account and user-defined dimensions to specify data to gather from planners. Use Entity to model the flow of planning information in the organization and establish the plan review path.

Aggregation Options

You can define calculations within dimension hierarchies using aggregation options. Aggregation options determine how child member values aggregate to parent members:

- + Addition
- - Subtraction
- * Multiplication
- / Division
- % Percent
- ~ Ignore
- Never (do not aggregate, regardless of hierarchy)

Storage Options

Table 12-2 Storage Options

Option	Impact
Dynamic Calc and Store	Calculates data values of members, and stores values.
Store	Stores data values of members.
Dynamic Calc	Calculates data values of members, and disregards the values.
Never Share	Prohibits members in the same dimension from sharing data values.
Shared	Allows members in the same dimension to share data values.
Label Only	Has no data associated with the member.

About Dynamic Calc

With dynamically calculated members, Oracle Hyperion Planning calculates data values of members, and disregards these values. The limit is 100 children under a Dynamic Calc parent. Changing a member's storage to Dynamic Calc may result in loss of data, depending on how the data was originally derived. You may need to update outlines, calculations, or both to get the dynamically calculated value.

Dynamic Calc Versus Dynamic Calc and Store

In most cases, you can optimize calculations and lower disk usage by using Dynamic Calc instead of Dynamic Calc and Store when calculating members of sparse dimensions. Use Dynamic Calc and Store for members of sparse dimensions with complex formulas, or that users retrieve frequently.

For members of dense dimensions, use Dynamic Calc. Dynamic Calc and Store provides only a small decrease in retrieval time and regular calculation time, and does not significantly lower disk usage. For data values accessed concurrently by many users, use Dynamic Calc. Retrieval time may be significantly lower than for Dynamic Calc and Store.

Note:

- Do not use Dynamic Calc for base-level members for which users enter data.
- Do not use Dynamic Calc for a parent member if you enter data for that member in a target version. Parent members set to Dynamic Calc are read-only in target versions.
- Data values are not saved for Dynamic Calc members.

About Store Data Storage

Do not set parent members to Store if their children are set to Dynamic Calc. With this combination, new totals for parents are not calculated when users save and refresh forms.

About Shared Data Storage

Use Shared to allow alternate rollup structures in the application.

About Never Share Data Storage

The default data storage type is Never Share when you add user-defined custom dimensions. You can use Never Share for parent members with only one child member that aggregates to the parent, to apply access to the child member.

About Label Only Data Storage

Label-only members are virtual members; they are typically used for navigation and have no associated data. Note:

- You cannot assign level 0 members as label-only.
- Label-only members can display values.
- Making dimension members label-only minimizes database space by decreasing block size.
- You cannot assign attributes to label-only members.

- In a multicurrency application, you cannot apply label-only storage to members of these dimensions: Entity, Versions, Currencies, and user-defined custom dimensions. To store exchange rates, use Never Share.
- Data Storage for children of label-only parents is set to Never Share by default.

 **Caution:**

Do not design forms in which label-only parents follow their first child member, as you cannot save data in the first child member. Instead, create forms with label-only parents selected before their children, or do not select label-only parents for forms.

About Entities

Entities typically match your organization's structure, such as geographical regions, departments, or divisions. Create entity members for groups that submit plans for approval. Entity members help define budget review, or approvals (see [Managing the Budgeting Process](#)).

For example, you may have regional centers preparing budgets for country headquarters. The country headquarters may prepare plans for corporate headquarters. To match this structure, create members for the regions, countries and headquarters. Specify regions as children of country members, and country members as children of headquarters.

Forms support multiple currencies per entity, enabling data entry for multiple currencies and reporting against one currency. However, Oracle Hyperion Planning supports a base entity for each entity. You can set the currency for entered values, which are converted to other currencies having defined exchange rates.

Base Currency

For a multicurrency application, specify each entity member's base currency. The default base currency for entity members is the currency specified when creating the application. For example, if U.S. Dollars is the default currency, you may specify Yen as the base currency for the Japan entity and U.S. Dollars for the United States entity. When using forms having values for the Japan entity, if the display currency is set to U.S. Dollars, values are converted to U.S. Dollars using the rates in the exchange rate table (assuming Yen is the local currency and U.S. Dollars is the reporting currency).

About Accounts

Account dimension members specify the information needed from budget planners. Create an account structure that lets budget preparers input data for budget items. You can define calculations in the account structure.

Account Types

Account type defines accounts' time balance (how values flow over time) and determines accounts' sign behavior for variance reporting with member formulas.

Examples of Using Account Types

Table 12-3 Using Account Types

Account Type	Purpose
Expense	Cost of doing business
Revenue	Source of income
Asset	Company resource
Liability and Equity	Residual interest or obligation to creditors
Saved assumption	Centralized planning assumptions ensuring consistency across the application

Summary of Account Types

Table 12-4 Summary of Account Types

Account Type	Time Balance	Variance Reporting
Revenue	Flow	Non-Expense
Expense	Flow	Expense
Asset	Balance	Non-Expense
Liability	Balance	Non-Expense
Equity	Balance	Non-Expense
Saved Assumption	User-defined	User-defined

Variance reporting and time balance settings are system-defined; only Saved Assumption is user-defined.

Time Balance Property

Time balance specifies how Oracle Hyperion Planning calculates the value of summary time periods.

Table 12-5 Time Balance Properties

Time Balance Property	Description	Example
Flow	Aggregate of all values for a summary time period as a period total.	Jan: 10 Feb: 15 Mar: 20 Q1: 45
First	Beginning value in a summary time period as the period total.	Jan: 10 Feb: 15 Mar: 20 Q1: 10
Balance	Ending value in a summary time period as the period total.	Jan: 10 Feb: 15 Mar: 20 Q1: 20
Average	Average for all the child values in a summary time period as the period total.	Jan: 10 Feb: 15 Mar: 20 Q1: 15

Table 12-5 (Cont.) Time Balance Properties

Time Balance Property	Description	Example
Fill	The value set at the parent is filled into all its descendents. If a child value changes, the default aggregation logic applies up to its parent. Consolidation operators and member formulas overwrite Fill values when the members are recalculated.	Jan: 10; Feb: 10; Mar: 10; Q1: 30
Weighted Average - Actual_Actual	Weighted daily average, based on the actual number of days in a year; accounts for leap year, in which February has 29 days. In the example, the average for Q1 is calculated: (1) Multiply each month's value in Q1 by the number of days in the month, (2) Sum these values, (3) Divide the total by the number of days in Q1. Assuming it is a leap year, the result is calculated: $(10 * 31 + 15 * 29 + 20 * 31) / 91 = 15$	Jan: 10 Feb: 15 Mar: 20 Q1: 15
Weighted Average - Actual_365	Weighted daily average, based on 365 days in a year, assuming that February has 28 days; does not account for leap years. In the example, the average for Q1 is calculated: (1) Multiply each month's value in Q1 by the number of days in the month, (2) Sum these values, (3) Divide the total by the number of days in Q1. Assuming it is not a leap year, the result is calculated: $(10 * 31 + 15 * 28 + 20 * 31) / 90 = 15$	Jan: 10 Feb: 15 Mar: 20 Q1: 15

You can use the Weighted Average - Actual_Actual and Weighted Average - Actual_365 time balance properties only with a standard monthly calendar that rolls up to four quarters. For information on how Planning calculates and spreads data with the different Time Balance settings, see *Oracle Hyperion Planning User's Guide*.

Account Types and Variance Reporting

An account's variance reporting property determines whether it is treated as an expense when used in member formulas:

- Expense: The actual value is subtracted from the budgeted value to determine the variance
- Non-Expense: The budgeted value is subtracted from the actual value to determine the variance

Setting Account Calculations for Zeros and Missing Values

With time balance properties First, Balance, and Average, specify how database calculations treat zeros and missing values with the Skip options.

Table 12-6 Effect of Skip Options When Time Balance is Set to First

Skip Option	Description	Example
None	Zeros and #MISSING values are considered when calculating parent values (the default). In the example, the value of the first child (Jan) is 0, and zeros are considered when calculating the parent value, so Q1 = 0.	Jan: 0 Feb: 20 Mar: 25 Q1: 0
Missing	Excludes #MISSING values when calculating parent values. In the example, the value of the first child (Jan) is #MISSING, and #MISSING values are not considered when calculating parent values, so Q1 = second child (Feb), or 20.	Jan: #MISSING Feb: 20 Mar: 25 Q1: 20
Zeros	Excludes zero values when calculating parent values. In the example, the value of the first child (Jan) is 0, and zero values are not considered when calculating parent values, so Q1 = the second child (Feb), or 20.	Jan: 0 Feb: 20 Mar: 25 Q1: 20
Missing and Zeros	Excludes #MISSING and zero values when calculating parent values. In the example, the value of the first child (Jan) is zero, and the value of the second child (Feb) is missing. Because missing and zero values are not considered when calculating parent values, Q1 = the third child (Mar), or 25.	Jan: 0 Feb: #MISSING Mar: 25 Q1: 25

Saved Assumptions

Use saved assumptions to centralize planning assumptions, identifying key business drivers and ensuring application consistency. You select time balance and variance reporting properties.

- Variance reporting determines the variance between budgeted and actual data, as an expense or non-expense.
- Time balance determines the ending value for summary time periods.

Examples of how time balance and variance reporting properties are used with saved assumption account members:

- Create a saved assumption of an expense type for variance reporting, assuming that the actual amount spent on headcount is less than the amount budgeted. To determine the variance, Oracle Hyperion Planning subtracts the actual amount from the budgeted amount.
- Determine the value for office floor space by using the time period's last value.
- Make an assumption about the number of product units sold at the end of the time period. Determine the final value for the summary time period by aggregating the number of units sold across time periods.

Data Type and Exchange Rate Type

Data type and exchange rate type determine how values are stored in account members, and the exchange rates used to calculate values. Available data type for account members' values:

- Currency - Stores and displays in the default currency.
- Non-currency - Stores and displays as a numeric value.
- Percentage - Stores a numeric value and displays as a percent.
- Date - Displays as a date.
- Text - Displays as text.

For accounts with the Currency data type, available Exchange Rate types (valid for any time period):

- Average - Average exchange rate
- Ending - Ending exchange rate
- Historical - Exchange rate in effect when, for example, earnings for a Retained Earnings account were earned or assets for a Fixed Assets account were purchased.

Accounts, Entities, and Plan Types

By assigning plan types for Entity and Account members, you set to which plan types the members' children have access. For example, Total Sales Account may be valid for Revenue and P&L, but Fixed Assets Account may be valid for only Balance Sheet. Not assigning a plan type to a member prevents that member's children from accessing that plan type.

When moving members, if the new parent is valid for different plan types, members remain valid only for plan types they have in common with the new parent. If the new parent of an account member has another source plan type, the member's source plan type is set to the first new valid plan type of that member.

Entities and Plan Types

Typically entity members prepare different plans. When defining entity members, specify plan types for which they are valid. Because forms are associated with plan types, you can control which entity members can enter data for each plan type.

Accounts and Plan Types

If accounts are valid for multiple plan types, specify the source plan type to determine which plan type's database stores the account value for them.

About User-Defined Custom Dimensions

You can add up to 13 user-defined custom dimensions. For example, you could add a dimension called Project to budget operating expenses for each project. You define properties, including name, alias, plan type, security, attributes, and attribute values.

⚠ Caution:

You cannot delete custom dimensions after you create them.

User-defined custom dimensions differ from the Entity and Account dimensions in that you assign valid plan types at the dimension level, not at the member level. All members of a user-defined custom dimension are valid for plan types assigned at the dimension level.



Adding or Editing User-Defined Custom Dimensions

User-defined custom dimensions must conform to guidelines listed in [Naming Restrictions](#).

Table 12-7 Properties for User-Defined Custom Dimensions

Property	Value
Dimension	Enter a name that is unique across all dimensions.
Alias	Optional: Select an alias table. Enter a unique alternate name for the dimension.
Description	Optional: Enter a description.
Valid for Plan Types	Select plan types for which the dimension is valid. Clearing this option makes all members of the dimension invalid for the deselected plan type.
Apply Security	Allow security to be set on the dimension members; must be selected before assigning access rights to dimension members. Otherwise, dimensions have no security and users can access members without restriction.
Data Storage	Select a data storage option. The default is Never Share.

To add or change user-defined dimensions:

1. Select **Administration**, then **Manage**, then **Dimensions**.
2. Click  or select an existing dimension and click .
3. Specify any of the properties listed above.
4. Click **Save**.
5. Click **OK**.

Click **Refresh** to revert to the previous values and keep the page open.

Setting Dimension Properties

Dimension properties must conform to guidelines listed in [Naming Restrictions](#).

Table 12-8 Dimension Properties

Property	Value
Dimension	Enter a dimension name.



Table 12-8 (Cont.) Dimension Properties

Property	Value
Alias	Optional: Select an alias table and enter an alternate name of up to 80 characters. Follow the dimension naming restrictions.
Description	Optional: Enter a description.
Valid for Plan Types	Select plan types for which the dimension is valid. Not available for Entity or Account dimensions.
Apply Security	Allow security to be set on dimension members. If you do not select this option, there is no security on the dimension, and users can access its members without restriction. Must be selected before assigning access rights to dimension members.
Data Storage	Select data storage options.
Display Option	Set application default display options for the Member Selection dialog box. Select Member Name or Alias to display members or aliases. Member Name:Alias displays members on the left and aliases on the right. Alias:Member Name displays aliases on the left and members on the right.
Enable custom attribute display	Display available and selected attributes for dimensions with associated attributes. Enable custom attribute display for dimensions with attributes.

Setting Dimension Density and Order

The Performance Settings tab enables you to set dimensions as sparse or dense and set their order of precedence.

To manage performance settings:

1. Select **Administration**, then **Manage**, and then **Dimensions**.
2. Select **Performance Settings**.
3. For each dimension, set its **Density** as **Dense** or **Sparse**.
See [About Sparse and Dense Dimensions](#).
4. Set the order of precedence by selecting a dimension and clicking  or .
See [About Reordering Dimensions](#).





Setting the Evaluation Order

The **Evaluation Order** tab enables you to specify which data type prevails when a data intersection has conflicting data types. For example, if Account members are set to the Currency data type, and Product members are set to the Smart List data type, you can set whether the Currency or Smart List data type prevails at an intersection.



To set evaluation order:

1. Select **Administration**, then **Manage**, and then **Dimensions**.

2. Select **Evaluation Order**.
3. Select the plan type and click **Go**.
4. From **Available Dimensions**, select dimensions and move them to **Selected Dimensions**:

-  moves selected dimensions
-  moves all dimensions
-  removes selected dimensions
-  removes all dimensions

You need select only dimensions whose members have specific data types (that is, their data type is not "Unspecified"). The data type "Unspecified" does not conflict with another data type.

5. If you select multiple dimensions, set the order of precedence by clicking  or  .
6. Click **Save**.

Working with Members

You can assign access rights to members, rearrange the dimension member hierarchy, and share members of the Entity, Account, and user-defined custom dimensions.

About Dynamic Members

Dynamic members are members that users can create when working with business rules. They're sometimes referred to as "on the fly members". An administrator enables end users to create dynamic members under a parent member and must refresh the database to create the required placeholders in Oracle Essbase. In business rules with runtime prompts, users can then create members by entering desired member names in the runtime prompts. Subsequent database refreshes will rename used dynamic children to the names specified by the end users and recreate the required placeholders in Essbase. For information about working with business rules and dynamic members, see *Oracle Hyperion Calculation Manager Designer's Guide*.

If a parent member is enabled for adding dynamic children (as described in this section), users can create new members by entering their name in the runtime prompt.

To enable a parent member for adding dynamic children:

1. Edit the parent member and select the option **Enable for Dynamic Children** (see [Adding or Editing Members](#)).
 - **Optional:** Set the member property **Number of Possible Dynamic Children** (the default is 10). This setting determines the number of placeholders that are created for dynamically adding or loading members under the parent. If all placeholders are used, subsequently added children are added as normal members and cannot be used until the database is refreshed.
 - **Optional:** Set the member property **Access Granted to Member Creator** (the default is Inherit).
2. Refresh the database to create the placeholders for dynamic members in the databases for the plan types in which the members are used.
3. In Oracle Hyperion Calculation Manager:

- a. Create a business rule with a runtime prompt (of the Variable type Member). Under the **Dynamic Member Parent** column, use the **Member Selector** to select the parent member that you enabled for dynamic children in Oracle Hyperion Planning.
- b. Select the option **Create dynamic members**.
- c. Deploy the business rule.

For information about working with business rules, see the *Oracle Hyperion Calculation Manager Designer's Guide*.

Notes:

- In Calculation Manager, selecting both the **Create dynamic members** and the **Delete dynamic members** options enables temporary dynamic members to be created for calculations, and those temporary dynamic members will be deleted after the business rule completes.
- Selecting the **Delete dynamic members** option alone presents the end user with the **Member Selector** in the runtime prompt so he can delete any member that he created dynamically under the parent (if he has write access to that member). This enables the end user to have full control over cleaning up and managing the members under the parent. The key is proper design to fit your requirements, setting the appropriate access rights to dynamic children with the **Access Granted to Member Creator** member property.

If all the above conditions are met, when users run the business rule with the runtime prompt, they enter the name of the dynamic member, and then click **Launch**. If the business rule runs successfully, the member is created in the dimension hierarchy under the dynamic member's parent.

Child members that you import under parent members enabled for dynamic children are added as dynamic child members if there are dynamic member placeholders in Essbase. After the placeholders are full, any remaining children are added as normal members, and cannot be used until the database is refreshed.



Note:

If you load a parent member that is enabled for dynamic children and its child members during the same import, the child members are loaded as normal members. This is because the database must be refreshed to create the placeholders in Essbase.

Adding or Editing Members

Members must conform to guidelines listed in [Naming Restrictions](#). Shared members must be consistent with [Working with Shared Members](#).

Table 12-9 Member Properties

Property	Value
Name	Enter a name that is unique across all dimension members.
Description	Optional: Enter a description.

Table 12-9 (Cont.) Member Properties

Property	Value
Alias Table	Optional: Select the alias table to store the alias name. Enter an alternate name for the member.
For Account members only: Account Type	Select the account type.
For Account members only: Variance Reporting	If the account type is Saved Assumptions, for Variance Reporting, select Expense or Non-Expense. Designate the saved assumption as a revenue, asset, liability, or equity account.
For Account members only: Time Balance	For Time Balance, select Flow, First, Balance, Average, Fill, Weighted Average - Actual_Actual, or Weighted Average - Actual_365. For descriptions, see Time Balance Property .
For Account members only: Skip	If the account type is Asset, Equity and Liability, select None, Missing, Zeros, or Missing and Zeros. For descriptions, see Setting Account Calculations for Zeros and Missing Values .
For Account members only: Exchange Rate Type	For Exchange Rate Type, select an option.
For Account members only: Data Type	For Data Type, select a data type.
Distribution	Sets the weekly distribution. Available for leaf Account members if the option was selected when creating the application and the base time period is 12 months.

Table 12-9 (Cont.) Member Properties


Property	Value
Hierarchy Type	Hierarchy Type is available for dimensions bound to an aggregate storage plan type. Aggregate storage dimensions are automatically enabled to support multiple hierarchies. The first hierarchy in a multiple hierarchy dimension must be stored.
	<div style="border: 1px solid #0070C0; padding: 10px; background-color: #E6F2FF;"> <p> Note:</p> <p>For members with a stored hierarchy type, the only valid plan type aggregation options are Addition or Ignore. In a stored hierarchy, the first member must be set to Addition. For members with a dynamic hierarchy type, all plan type aggregation options are valid. Stored hierarchy members that are not children of Label Only members must have Addition set as the consolidation operator. Children of Label Only members can be set to Ignore.</p> </div>
Data Storage	Select a data storage property. The default is Never Share for new custom dimension members (except root members).
Two Pass Calculation	Recalculate values of members based on values of parent members or other members. Available for Account and Entity members with Dynamic Calc or Dynamic Calc and Store properties.
For Entity members only: Base Currency	Select the Entity member's base currency.
Display Option	Set application default display options for the Member Selection dialog box. Select Member Name or Alias to display members or aliases. Member Name:Alias displays members on the left and aliases on the right. Alias:Member Name displays aliases on the left and members on the right.

Table 12-9 (Cont.) Member Properties

Property	Value
Plan Type	Select plan types for which the member is valid.

 **Note:**

A member can belong to both aggregate storage and block storage plan types.

Select an aggregation option for each selected plan type. You can select a source plan only if multiple plan types are valid for the member. Only plan types and aggregation options for which the member's parent is valid are available. If the parent is not valid for a plan type or aggregation option, neither is the child member. Deselecting a plan type for an account or entity parent member deselects it for all descendents of that parent. For members with a stored hierarchy type, the only valid aggregation options are Addition or Ignore.

 **Caution:**

Deselecting a plan type for dimension members after data is entered into an application may result in loss of data when an application is refreshed. For account members, data is lost if the deselected plan type is the source plan type.

Members of a custom dimension can set usage by plan type, similar to the Account and Entity dimensions.

For Entity members only: Base Currency
For Account members only: Source Plan Type

Select the base currency for the Entity member.

Select the source plan type for the member. A shared member is a pointer to the base member and is not stored; this is disabled for shared members. The Source Plan Type of a shared Account member matches the source plan type of the base member, even though Source Plan field is unavailable because it does not apply to shared members.

Table 12-9 (Cont.) Member Properties

Property	Value
Smart Lists	Optional: Select a Smart List to associate with the member.
Enable for Dynamic Children	Enables users to create children for this member by entering a member name in the runtime prompt for a business rule that has been configured with a dynamic parent member (see About Dynamic Members).
Number of Possible Dynamic Children	This option is available if Enable for Dynamic Children is selected. Enter the maximum number of dynamically-added members that users can create. The default is 10.
Access Granted to Member Creator	This option is available if Enable for Dynamic Children is selected. Determines the access that member creators have to dynamic members that they create with a runtime prompt: <ul style="list-style-type: none"> • Inherit—The member creator will inherit the closest parent's access to the newly-created member. • None—The member creator will not be assigned any access to the newly-created member. (An administrator can later assign the member creator access to the members.) • Read—The member creator will be assigned Read access to the newly-created member. • Write—The member creator will be assigned Write access to the newly-created member.

 **Note:**

If an administrator changes these settings, they affect only future dynamic members; they do not retroactively affect dynamic members.

To add or edit members:

1. Select **Administration**, then **Manage**, and then **Dimensions**.
2. Select the dimension.
3. Perform one action:
 - To add a child member, select the parent level of the dimension hierarchy to which to add a member and click **Add Child**.
 - To add a sibling, select the level of the dimension hierarchy to which to add a sibling and click **Add Sibling**.

- To edit a member, select that member from the dimension hierarchy and press **Enter** or click **Edit**.

 **Note:**

To add an All Years parent member that includes all year members, select the Years dimension and then click **All Years**. The All Years parent member enables users to view the accumulated data across multiple years, for example, a project's total cost up to its end date. The All Years member does not include the No Year member, if one is defined for the application.

4. On **Member Properties**, set or change member properties described in the table above. If you do not see the new member on the page, click **Next**.
5. Click **Save** to save information to the relational database and see changes in the dimension hierarchy.
6. Refresh the database so edited members are visible to planners entering data.
7. After creating a dimension member, you typically complete these tasks:
 - Assign access. See [Assigning Access to Members and Business Rules](#).
 - Specify attributes.

Deleting Members

Each data value is identified by a set of dimension member values and a plan type. Deleting dimension members or deselecting the plan type results in data loss when refreshing an application. Deleting entity members deletes all planning units (including data) associated with them.

 **Caution:**

Before beginning this procedure, perform a backup. See [Backing Up Applications and Application Databases](#) and the *Oracle Enterprise Performance Management System Backup and Recovery Guide*.

Before deleting members, understand where in the application they are used (in which forms, planning units, exchange rates, and so on) by using Show Usage.

You must delete the entity member throughout Oracle Hyperion Planning before deleting it from Dimensions. For example, if the entity member is used in a form, you must delete it from the form before deleting it from Dimensions.

When deleting a large subtree of entities, you can improve performance if you first exclude planning units for the subtree (by excluding the root member) for all scenarios and versions. See [Review Process](#).

To delete members:

1. Select **Administration**, then **Manage**, then **Dimensions**.
2. Select the dimension whose member you want to delete.
3. From the dimension hierarchy, select the entity member to delete.

4. Click **Delete**.
Deleting a base member also deletes its shared members.
5. Click **OK**.
6. Update and validate business rules and reports.

Deleting Parent Members

Data values are identified by a set of dimension member values and a plan type. Deleting dimension members or deselecting the plan type results in data loss when refreshing the application.

Caution:

Before beginning this procedure, perform a backup. See [Backing Up Applications and Application Databases](#) and the *Oracle Enterprise Performance Management System Backup and Recovery Guide*.

To delete a parent member and all its descendants from the dimension hierarchy:

1. Select **Administration**, then **Manage**, then **Dimensions**.
2. Select the dimension whose member and descendants to delete.
3. Select the member whose branch to delete.
4. Click **Delete**.
5. Click **OK**.

Viewing Member Properties from Forms

To view member properties from forms:

1. In the form, select a row or column member and right-click.
2. Select **Show member in outline**.
The Dimensions page displays the member highlighted in the hierarchy.
3. **Optional:** Select **Edit** to view the member's properties, then click **Cancel**.

Working with Shared Members

Sharing members allow alternate rollup structures within a Oracle Hyperion Planning application. A base member must exist before you can create a shared member. You can create multiple shared members for the base member. A base member must display before its shared members in position from top to bottom.

Shared members are available for Entity, Account, and user-defined custom dimensions. Shared member values can be ignored to avoid double-counting values when you roll up the outline.

Shared members share some property definitions with base members, such as member name, alias name, base currency, and plan types for which members are valid. Shared members must have unique parent members and different rollup aggregation settings. Custom attributes,

custom attribute values, and member formulas are not allowed for shared members. Renaming base members renames all shared members.

Shared members cannot be moved to another parent member. You must delete shared members and recreate them under different parent members. Shared members must be at the lowest level (level zero) in the hierarchy and cannot have children. The base member need not be level zero. You can enter data in shared members, and values are stored with base members.

Shared members are displayed similarly to base members in the dimension hierarchy for member selection in Oracle Smart View for Office.

Creating Shared Members

You create shared members the same way as other members, with these differences:

- The base member cannot be the parent of the shared member.
- You cannot add a shared member as a sibling to the base member.
- You must give the shared member the same name as its base member. It can have a different description.
- You must select Shared as the Data Storage for the shared member.

Working with Attributes

Use attributes to group members using the same criterion. You can assign attributes to sparse dimensions only. You cannot assign attributes to label-only members. Attribute dimensions do not have aggregation properties because parents are dynamically calculated.

The Account dimension is usually defined as dense, so you cannot assign attributes to it unless it is changed to sparse for all plan types. If you change a dimension from sparse to dense, all attributes and attribute values for that dimension are automatically deleted.

Attributes can have data types of text, date, Boolean, and numeric, as described in [Understanding Attribute Data Types](#). Attribute names must conform to guidelines listed in [Naming Restrictions](#). When attributes are defined, you can use the Member Selection dialog box to select attribute functions, such as Equal and GreaterOrEqual.

To create and change attributes, attribute values, and aliases:

1. Select **Administration**, then **Manage**, then **Dimensions**.
2. Select a sparse dimension for which to define an attribute, attribute value, or alias.
Only sparse dimensions can contain attributes.
3. Select the top level in the dimension hierarchy, and click **Edit**.
4. In the Dimension Properties dialog box, click **Custom Attributes**.
If the dimension is not sparse, **Custom Attributes** is not available.
5. Select options.
 - To create attributes, click **Create**. Type an attribute name, and select a data type: **Text**, **Date**, **Boolean**, or **Numeric**. See [Understanding Attribute Data Types](#). You cannot modify the data type after the attribute is created.

Plan type options are available for Entity dimension attributes only. You cannot change this setting after the attribute is created.
 - To modify attributes, click **Modify**, and update the attribute name.

- To set aliases for attributes, select an attribute and an attribute value, click **Alias**. Select an alias table, type an alias name, and click **Close**.
6. Click **Close**.

When you click **Close**, the hierarchy is validated and an error displays if issues are detected. For example, date attribute values must be entered in the correct format, and numeric and date attribute dimensions must have at least one attribute value defined.
 7. Update and validate business rules and reports.

Understanding Attribute Data Types

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

- Text attributes enable basic attribute member selection and attribute comparisons in calculations. When you perform such comparisons, characters are compared. For example, a package type Bottle is less than a package type Can because B precedes C in the alphabet.
- Numeric attribute dimensions use numeric values for the names of level 0 members. You can include the names (values) of numeric attribute dimension members in calculations. For example, you can use the number of ounces specified in an Ounces attribute to calculate profit per ounce for each product. You can also associate numeric attributes with ranges of base dimension values, for example, to analyze product sales by market population groupings.
- Boolean attribute dimensions in a database contain only two members. When a Boolean attribute dimension is added in Planning, two attribute values, True and False, are created for this attribute dimension by default. A base dimension, such as Account or Entity, can be associated with only one attribute dimension that has the Boolean data type.
- Date attributes can specify the date format as month-day-year or day-month-year, and sequence information accordingly. You can use date attributes in calculations, for example, comparing dates in a calculation that selects product sales since 12-22-1998. Users can set the date format by selecting an option in Attribute Dimension Date Format in Application Settings preferences.

For detailed information about attributes and attribute values, see the *Oracle Essbase Database Administrator's Guide*.

Deleting Attributes

When you delete an attribute, all attribute values associated with the attribute are also deleted. Attribute values are removed from members to which they had been assigned, and the attribute is removed from dimensions to which it was assigned.

To delete attributes:

1. Select **Administration**, then **Manage**, then **Dimensions**.
2. Select the sparse dimension for which to delete an attribute, and click **Edit**.
3. Click **Custom Attributes**.
4. Select the attribute to delete.
5. Above the **Attributes** column, click **Delete**.
6. Click **OK**.

7. Update and validate business rules and reports.

Working with Attribute Values

Attribute values provide users with another way of selecting dimensions members when using forms. Data values for attribute values are dynamically calculated but not stored.

Creating Attribute Values

You can define attribute values for sparse dimensions, which are typically the Entity and user-defined custom dimensions. After you define an attribute value for a dimension, you can assign it to members of that dimension.

To create attribute values:

1. Select **Administration**, then **Manage**, then **Dimensions**.
2. Select the sparse dimension for which to create an attribute value.
3. Select the top level in the dimension hierarchy.
4. Click **Custom Attributes**.
5. On the **Manage Attributes and Values** page, select the attribute for which to specify a value.
6. Above the **Attribute Values** column, click **Create**. If the options are available, you can click **Add Child** or **Add Sibling**.
7. On **Create Attribute Value**, in **Name**, enter a name.
8. Press **Enter** or **Save**.
9. Click **Cancel**.




Assigning Attribute Values to Members

You can assign attribute values members of a dimension that are defined as sparse for all plan types. Attribute values must be assigned to the same-level sparse dimension members. Otherwise, errors display during refresh.

To assign attribute values to members:

1. Select **Administration**, then **Manage**, then **Dimensions**.
2. Select the sparse dimension for whose member you want to assign an attribute value.
3. In the **Dimension** hierarchy, select a member to which to assign an attribute value.
4. Click **Edit**.

For members assigned attribute values: Click **View** to change a member's attribute value.

5. Select **Attribute Values**.
6. Select attribute values to assign to the member.
7. Perform an action:
 - To assign the value to the selected member, click .
 - To remove a value from the selected member, select the value to remove and click .
 - To remove all values from the selected member, click .

8. Click **Save**.

Modifying Attribute Values

To modify attribute values:

1. Select **Administration**, then **Manage**, then **Dimensions**.
2. Select the sparse dimension for which to modify an attribute value.
3. Select the top level in the dimension hierarchy.
4. Click **Custom Attributes**.
5. For **Attributes**, select the attribute containing the value to modify.
6. For **Attribute Values**, select the attribute value.
7. Above **Attribute Values**, click **Modify**.
8. On **Modify Attribute Value**, in **Name**, enter a name.
9. Click **Save**.

Deleting Attribute Values

When you delete an attribute value, it is removed from custom dimension members to which it is assigned.

To delete attribute values:

1. Select **Administration**, then **Manage**, then **Dimensions**.
2. Select the sparse dimension containing the attribute for which to delete a value.
3. Select the top level in the dimension hierarchy.
4. Click **Custom Attributes**.
5. For **Attributes**, select the attribute containing attribute values to delete.
6. For **Attribute Values**, select attribute values to delete.
To select all attribute values for deletion, select **Attribute Values**.
7. Above the **Attribute Values** column, click **Delete**.
8. Click **OK**.
9. Update and validate business rules and reports.

Customizing Calendars

Use the Period dimension to work with the yearly calendar rollup structure. When creating the application, the administrator specifies the base time periods that span the Oracle Hyperion Planning database. Use the Years dimension to add years to the calendar.

Table 12-10 Calendar Tasks

Task	See Topic
Define how years roll up.	Defining How Calendars Roll Up.
Create and edit Summary Time Periods.	Creating and Editing Summary Time Periods.

Table 12-10 (Cont.) Calendar Tasks

Task	See Topic
Delete Summary Time Periods.	Deleting Summary Time Periods.
Work with the fiscal year and period hierarchy.	Working with the Years Dimension.

Defining How Calendars Roll Up

Table 12-11 Calendar Roll Up

Base Time Period	Roll Up
12 Months	Four quarters are created per year. Months roll up into parent quarters and quarters roll up into years.
Quarters	Quarters roll up into years.
Custom	No default rollup structures. A flat list of the custom base time periods displays.

After the application calendar is created, you cannot change the base time period or reduce the number of years in the calendar. Administrators can change the names, descriptions, aliases, and ranges of the summary time periods in the hierarchy.

Creating and Editing Summary Time Periods

You can change such aspects as name, description, alias, starting period, and ending period. However, you cannot change the order of base time periods or skip base time periods. The range cannot extend beyond the current fiscal year.

You must work from the top of the hierarchy to the bottom when creating summary time periods. (Otherwise, Oracle Hyperion Planning views the rollup structure as asymmetrical and you cannot continue.) The summary time period displays in the hierarchy as a parent of the selected item. To enforce a balanced hierarchy, all base members must be the same number of levels from the root.

To create or edit summary time periods:

1. Select **Administration**, then **Manage**, then **Dimensions**.
2. Select **Period**.
3. Perform one action:
 - To add a time period, select the level in the dimension hierarchy above which to add, and click **Add**.
 - To edit a time period, select the time period and click **Edit**.
4. For **Name**, enter or change the name for the summary time period.
5. **Optional:** For **Description**, enter a description.
6. **Optional:** For **Alias**, select an alias table to use. Enter an alias.
The default table is used if you do not select one.
7. For **Start Period**, select the starting period.

The range cannot extend beyond the current fiscal year. For summary time periods, **Start Period** displays the first child, or all children except the first child of the sibling above it.

8. For **End Period**, select the ending period.

For summary time periods, **End Period** displays the last child, or all children from the Start Period through the next sibling's children, except the last child.

9. Click **Save**.

Deleting Summary Time Periods

When you remove a summary time period from the hierarchy, its children are moved into another summary time period:

- If you delete the first summary time period, children are moved into the next sibling of the summary time period.
- If you delete the last summary time period, children are moved into the previous sibling of the summary time period.
- If you delete a summary time period from the middle of a hierarchy, children are moved into the previous sibling of the summary time period.

To delete summary time periods:

1. Select **Administration**, then **Manage**, then **Dimensions**.
2. Select **Period**.
3. For **Period hierarchy**, select the summary time period to delete.
You cannot delete base time periods.
4. Click **Delete**.
5. Click **OK**.

Working with the Years Dimension

Use the Years dimension to work with calendar years.

Table 12-12 Years Tasks

Task	See Topic
Add years to the calendar.	Adding Years to the Calendar .
Add or update the description and alias for a year.	Editing Year Information .
Set the fiscal year and work with years.	Setting the Fiscal Year .
Add an All Years parent member that includes all Years members (except No Year, if that member exists).	Adding or Editing Members

Adding Years to the Calendar

You can add years to the calendar, but you cannot reduce the number of calendar years without creating a database.

To add years to the calendar:

1. Select **Administration**, then **Manage**, then **Dimensions**.

2. Select **Years**.
3. Click **Add Years**.
4. For **Number of Years to Add**, enter the number of years to add to the calendar.
5. Click **Add Years**.

 **Note:**

To add an All Years parent member that includes all years members, click **All Years**. The All Years parent member enables users to view the accumulated data across multiple years, for example, a project's total cost up to its end date. (This parent member does not include the No Year member, if one exists.)

Setting the Fiscal Year

You can change the time period or current year.

To change the current year or time period:

1. Select **Administration**, then **Manage**, then **Dimensions**.
2. Select **Years** or **Period**.
3. Click **Options**.
4. On **Set Current Period and Year**, from **Current Year**, select the current year.

Current Month and Current Year set defaults for the Month and Year when scenarios are created. For example, if Current Year is set to FY08 and Current Month is set to Aug, when users create scenarios, these values display as defaults in the Start Yr, Start Period, End Yr, End Period fields.

5. For **Current Time Period**, select the current period.
6. Click **OK**.

Editing Year Information

You can add or update the description and alias for a year.

To edit years:

1. Select **Administration**, then **Manage**, then **Dimensions**.
2. Select **Years**.
3. Click **Edit**.
4. Enter a description for the year.
5. For **Alias Table**, select the alias table to use, and then enter an alias name.
6. Click **Save**.

Renaming Time Periods

You can rename root-level, base time periods, and user-defined summary time periods.

Assigning Aliases to Summary Time Periods

You can assign and change aliases to base time periods and summary time periods.

To assign or change the alias:

1. Select **Administration**, then **Manage**, then **Dimensions**.
2. Select **Period**.
3. Select the summary time period.
4. Click **Edit**.
5. For **Alias Table**, select the alias table to use.
6. Enter an alias name.
7. Click **Save**.

Editing the BegBalance Member

You can edit the BegBalance member of the Period dimension. As the first time period in the application, the BegBalance member is useful for entering beginning data when you start a new application, fiscal year, or calendar year. You can rename and describe BegBalance and give it an alias.

To edit the BegBalance member:

1. Select **Administration**, then **Manage**, then **Dimensions**.
2. For **Dimensions**, select **Period**.
3. Select the first member, **BegBalance**.
4. Click **Edit**.
5. For **Edit Period**:
 - Enter a name.
 - Enter a description.
 - Select an alias table to use for the BegBalance member, and enter an alias.
6. Click **Save**.

Setting Up Currencies

You can plan, forecast, and analyze financial information in one or more currencies. You can create, edit, and delete currencies. Administrators control:

- Which currencies an application uses, including for reporting
- How currencies display in reports and forms
- How currencies convert to other currencies
- Whether a triangulation currency converts currencies
- When currency conversions occur

Enabling Multiple Currencies

If an application supports multiple currencies, you can enable multiple currencies per entity on forms. See [Setting Form Layout](#). When selecting business rules for forms, you can select the Calculate Currencies business rule to convert values among the available currencies. See [Selecting Business Rules](#).

Working with Multiple Currencies

If multiple currencies are enabled, users can see values converted from the local currency to a reporting currency and can override a cell's base currency.

Note:

- When the local currency is selected on forms, the default stored and displayed currency for cells is the entity's base currency (which you specify). Users can enter data values only into local currency members. If the local currency member is selected, all currencies specified for the application are available as input types.
- You can set dimension properties for each currency in the Edit Currency dialog box. In preferences, users can select different display options, and can select Currency Setting to apply the properties set by the administrator.
- Currencies can be converted only to reporting currencies. Users cannot enter data into cells displayed in reporting currencies. The application's main currency is by default a reporting currency. You can change which currencies are reporting currencies.
- You can load values into a reporting currency by using Oracle Essbase Adapter to populate values directly into Essbase.
- Currencies defined for the application are valid currencies for data entry. Valid currencies for data entry are displayed in a list that users access by clicking the Currency link during data entry.
- To get meaningful results, roll up values in one common reporting currency. If members of a subtotal have mixed currencies, the currency type is blank and the currency symbol does not display. For example, adding 10 US dollars and 10 Japanese yen to a value of 20 makes no sense.
- An application with 500 time periods can successfully run currency conversion calc scripts only if the time periods have default names, TP 1 through 500. Otherwise, the conversion calc script you try to create exceeds the 64K limit.
- User-defined currency conversion calc scripts created when a database is created or refreshed may be available in Oracle Smart View for Office, depending on user access. When using user-defined currency conversion calc scripts, Oracle recommends changing the order so the currency conversion calc script is first, before Calculate Form.
- Currency codes associated with input values are stored as numeric values. These codes are calculated in dimension formulas, calc scripts, and business rules. The calculated values of these currency codes may translate to currency codes that are incorrect or invalid. Where there are children with mixed currencies, review calculated results on the upper levels.
- If a parent has multiple children, of whom only one child has an overridden currency, the parent inherits the overridden currency code (which is not displayed on forms).
- In certain cases, parent entities display #MISSING when trying to convert to a selected currency. Ensure that a currency rate is entered for each combination of local currencies

and selected currencies on forms or reports. Currency combinations must exist for all mixed-currency children entities and parent members.

- Input of multiple currencies to one entity is not supported in Smart View. If worksheets include mixed currency types, users could inadvertently enter values in the wrong currency.

About the Calculate Currencies Business Rule

The Calculate Currencies business rule is based on the dimensions and members on the form. It converts data from the local currency to the reporting currency specified on the form, applying the exchange rate conversions. It:

- Does not calculate subtotals. To subtotal values, run the Calculate Form business rule (or a customized business rule that includes aggregation) after converting currencies.
- Ignores #MISSING values.
- Can be turned on or off by associating or disassociating it with forms during form design.
- Is set by default to not run when saving data.

Exchange Rate Types

These exchange rates are associated with currencies: Historical, Average, and Ending. The exchange rate type for each account is specified in the Member Property dialog box. For average and ending rate types, enter values for all time periods. For historical rate types, enter one rate value that is used for all time periods, including the Beginning Balance period. For the Beginning Balance period, enter one rate value used for that time period for average and ending rate types.

Oracle Hyperion Planning supports currency conversion by triangulation through a triangulation currency.

Scaling

You can specify scaling data values when displayed in certain currencies. For example, you can set the scaling for Yen to Thousands, then enter 10,000 as a value for the Japan entity on a form with the Local member selected for the Currency dimension. When you select Yen as the currency member for the form, the scaling is applied and 10 displays as the value for Japan.

Number Formatting

You can determine the initial display of numerical values for non-currency and currency data types in forms:

- Thousands separator:
 - None: 1000
 - Comma: 1,000
 - Dot: 1.000
 - Space: 1 000
- Decimal separator:
 - Dot: 1000.00
 - Comma: 1000,00

- Negative number sign:
 - Prefixed minus: -1000
 - Suffixed minus: 1000-
 - Parentheses: (1000)
- Negative number color:
 - Black
 - Red

Reporting Currencies

A reporting currency is the currency in which your company prepares financial statements. Oracle Hyperion Planning supports currency conversion from local currencies to one or more reporting currencies. Converted reporting currency values are stored and read-only for all users. An application's default currency is the default reporting currency. You can disable a currency as a reporting currency.

Checking How Currencies are Used

You can view how an application uses currency: whether a currency is the default, is used for triangulation currency or by an entity, or has a conversion or exchange relationship with other currencies.

To see how currencies are used:

1. Select **Administration**, then **Manage**, then **Dimensions**.
2. Select **Currencies**.
3. Select the currency for which you want information.
4. Click **Show Usage**.

Creating Currencies

Select from a predefined list or create your own. You can specify:

- The three-letter code
- The symbol
- A description of up to 256 characters
- The scaling factor to use when values are displayed
- The triangulation currency to use for currency conversion
- The alias table to use to display aliases
- Number formatting, including thousands separator, decimal separator, negative sign, and color
- Whether it is a reporting currency

To create currencies:

1. Select **Administration**, then **Manage**, then **Dimensions**.
2. Select **Currency**.
3. Click **Add**.

4. From **Create Currency**:
 - To add a predefined currency, select **Select standard Currencies**.
 - To create a currency, select **Create new currency** and specify properties:
 - For **Code**, enter an abbreviation or identifier of up to three characters.
 - **Optional**: For **Description**, enter a name, such as Japanese yen.
 - For **Symbol**, enter a symbol or select a symbol from the list.
 - **Optional**: For **Scale**, select how to enter and display the currency. For example, 3 yen represents 3000 yen if scaling is set to thousands.
 - **Optional**: For **Triangulation Currency**, select the currency to use as the common third currency for conversion.
 - **Optional**: For **Alias Table**, select the alias table to use.
 - **Optional**: For **Alias**, enter a name for the currency alias.
5. **Optional**: Select **Reporting Currency** (see [Working with Multiple Currencies](#)).
6. **Optional**: For **Thousands Separator**, select how to display the thousands separator (it must differ from the decimal separator).
7. **Optional**: For **Decimal Separator**, select how to display numbers with decimal values (it must differ from the thousands separator).
8. **Optional**: For **Negative Sign**, select how to display negative numbers:
 - **Prefixed minus**: -1000.
 - **Suffixed minus**: 1000-
 - **Parentheses**: (1000)
9. **Optional**: For **Negative Color**, select the display color.
10. **Optional**: Select the type of **Data Storage**.
11. **Optional**: Select **Two Pass Calculation**.
12. **Optional**: Select the **Data Type**.
13. **Optional**: Select a **Smart List**.
14. Click **Save**.

Editing Currencies

To edit currencies:

1. Select **Administration**, then **Manage**, and then **Dimensions**.
2. Select **Currency**.
3. Select the currency to edit.
4. Click **Edit**.
5. Modify properties:
 - To select from the predefined symbols, select one from the **Select from Predefined Symbols** drop-down list.
 - To change the currency's symbol, for **Symbol**, enter or select the symbol.
 - For **Scale**, set how to enter and display the currency.

- For set currency precision (the number of digits to the right of the decimal place), select a number from 1 to 10 from the **Precision** drop-down list.

None is the default.

 **Note:**

Administrators can override this setting for forms. See [Setting Form Precision and Other Options](#).

- To specify the currency as a reporting currency, select **Reporting Currency**. See [Working with Multiple Currencies](#).
 - For **Thousands Separator**, select how to display the thousands separator (it must differ from the decimal separator).
 - For **Decimal Separator**, select how to display numbers with decimal values (it must differ from the thousands separator).
 - For **Negative Sign**, select how to display negative numbers:
 - **Prefixed Minus:** -1000.
 - **Suffixed Minus:** 1000-
 - **Parentheses:** (1000)
 - **Use Default Setting:** Apply the display setting for the currency (see [Creating Currencies](#)).
 - For **Negative Color**, select the display color.
6. Click **Save**.

Deleting Currencies

You cannot delete the default currency.

To delete currencies:

1. Select **Administration**, then **Manage**, then **Dimensions**.
2. For **Dimension**, select **Currencies**.
3. For **Currency**, select the currency to delete.
4. Click **Show Usage** to determine if the currency is the default currency, a triangulation currency, or associated with an entity. You cannot delete a currency that meets these criteria.

If you delete a currency defined in the exchange rate table, it is deleted from the table.

5. Click **Close**, **OK**, **Delete**, and **OK**.
6. Update and validate business rules and reports.

Specifying Exchange Rates

Use exchange rates to convert values from one currency to another. You can:

- Enable budget preparers in various countries to create plans in other currencies
- Show summary report data in a currency

- Summarize values from multiple currencies into one currency

For example, you might specify yen as the base currency for the Japan entity and US dollars for the United States entity. When you display a form having values for the Japan entity and the form's display currency is set to US dollars, the exchange rates for the yen is used to convert the values for Japan to US dollars. If the display currency is set to yen, the exchange rates for US dollars converts values for the United States entity to yen.

To specify exchange rates, you must set up multiple currencies when creating an application.

About Exchange Rate Tables

Each application has a default currency specified when the application is created. When you specify exchange rate tables, only the default currency and triangulation currencies are available as destination currencies. You can enter exchange rates from source currencies to default or triangulation currencies.

You can create multiple exchange rate tables. Each table is typically associated with multiple scenarios, but each scenario can be associated with only one exchange rate table. When creating scenarios, select the exchange rate table for converting currencies.

Enter conversion values between the default currency and currencies defined in the Exchange Rates page. Exchange rate tables span all application time periods, so you can apply exchange rates to all scenarios. When creating or modifying exchange rate tables, you must refresh the application to store them in the plan types. See [Creating and Refreshing Application Databases](#).

Hsp_Rates Dimension

A multiple currency application includes the Hsp_Rates dimension for storing exchange rates. It includes these members and others that store currency rates:

- Hsp_InputValue: Stores data values
- Hsp_InputCurrency: Stores currency types for data values

When generating reports or loading data, refer to the Hsp_InputValue member. When loading data, you must load data against the local currency. You need not refer to the Hsp_InputCurrency member.

By default, the Hsp_Rates dimension is set to Sparse. You can change this (see [Setting Dimension Density and Order](#)).

Triangulation

Oracle Hyperion Planning supports currency conversion by triangulation through an interim currency called the triangulation currency. If you modify a currency's triangulation currency, you must re-enter exchange rates for the triangulation currency property and refresh the application to transfer and store the exchange rates. You cannot select the application's default currency as a triangulation currency.

Calculation Method

When you input exchange rates for converting between currencies, you can select Multiply or Divide as the calculation method. For example, if you select 1.5 as the rate for converting British Pounds to US dollars, and select multiply as the calculation method, 1 British Pound is converted to 1.5 US dollars.

Setting Up Scenarios

Each scenario/version combination contains data for accounts and other dimensions of each entity. After users enter data for an entity for a scenario and version, they can submit or promote the data for the entity to other users for review and approval.

Task	Topic
Create a scenario.	See Creating Scenarios .
Edit a scenario.	See Editing Scenarios .
Delete a scenario.	See Deleting Scenarios .
Copy a scenario.	See Copying Scenarios .
Delete supporting detail associated with a scenario.	See Deleting Supporting Detail Associated With a Scenario .

About Scenarios

Use scenarios to:

- Apply different planning methods.
- Create forecasts.
- Enter data into scenarios.
- Associate scenarios with different time periods or exchange rates.
- Assign user access rights by scenario.
- Report on scenarios.
- Compare and analyze scenarios.

You can group applications into multiple plans with individual review cycles. Scenarios can cover different time spans.

Time Periods

Assign each scenario a range of years and time periods, and specify the Beginning Balance time period. When users access forms, they can enter into that scenario only years and periods within the range. Years and periods outside of the range display as read-only. You can modify the time range.

Exchange Rate Table

If an application converts currencies, assign an exchange rate table to the scenario. By assigning different exchange rate tables to scenarios, you can model the effects of currency rate assumptions.

Access Permissions

Specify access permissions to Scenario dimension members for groups or users to determine who can view or modify data. A user or group can have only one of these access permissions: Read, Write, or None. Access permissions for a user can be combined based on groups to which the user belongs.

Creating Scenarios

To create scenarios:

1. Select **Administration**, then **Manage**, and then **Dimensions**.
2. Select **Scenarios**.
3. Click **Add Child**.
4. For **Scenario**, enter a name.
5. **Optional**: For **Description**, enter a description.
6. For **Start Yr.**, **Start Period**, **End Yr.**, and **End Period**, select the time period to associate with the scenario.
7. **Optional**: For **Exchange Rate Table**, select an exchange rate table to associate with the scenario.

If an application uses multiple currencies, associate a scenario with an exchange rate table to enable currency conversions.

8. **Optional**: For **Alias**, select an alias table to associate with the scenario, and enter a description.
9. **Optional**: Select **Include BegBal as Time Period** to include the BegBalance time period in this scenario for currency conversion.
10. **Optional**: Select **Enabled for Process Management** to include this scenario in approvals.
11. Click **Save**.

Editing Scenarios

To modify scenarios:

1. Select **Administration**, then **Manage**, then **Dimensions**.
2. For **Dimension**, select **Scenario**.
3. Select the scenario to edit.
4. Click **Edit**.
5. **Optional**: For **Scenario**, enter a name.
6. **Optional**: For **Description**, enter a description.
7. For **Start Yr.**, **Start Period**, **End Yr.**, and **End Period**, select the time period to associate with the scenario.
8. **Optional**: For **Exchange Rate Table**, select an exchange rate table to associate with the scenario.

If an application uses multiple currencies, associate a scenario with an exchange rate table to enable currency conversion.

9. **Optional**: For **Alias**, select an alias table to associate with the scenario, and enter the description.
10. **Optional**: Select **Include BegBal as Time Period** to include the BegBalance time period in this scenario for currency conversion.
11. **Optional**: Select **Enabled for Process Management** to use this scenario in approvals.

12. Click **Save**.

Deleting Scenarios

When you delete scenarios, all planning units that use the scenario (including data) are deleted. You cannot delete scenarios used in planning units that are started, or assigned to an axis on a form. You must first remove references to scenarios from forms and assign different scenarios.

To delete scenarios:

1. Select **Administration**, then **Manage**, then **Dimensions**.
2. Select **Scenarios**.
3. Select the scenarios to delete. At least one scenario must remain in the application.
4. Click **Delete**.
5. Click **OK**.
6. Update and validate business rules and reports.

Copying Scenarios

Only scenario properties are copied. Data values and access rights associated with the original scenario are not copied to the new scenario.

To copy scenarios:

1. Select **Administration**, then **Manage**, then **Dimensions**.
2. Select **Scenarios**.
3. Select the scenario to copy.
4. Click **Copy Scenario**.
5. For **Copy to Scenario**, enter a name.
6. **Optional**: For **Description**, enter a description.
7. For **Start Yr.**, **Start Period**, **End Yr.**, and **End Period**, select the time period to associate with the scenario.
8. **Optional**: For **Exchange Rate Table**, select an exchange rate table to associate with the scenario.

If an application uses multiple currencies, associate a scenario with an exchange rate table to enable currency conversion.
9. **Optional**: For **Alias Table**, select an alias table to associate with the scenario, and enter a description.
10. **Optional**: Select **Include BegBal as Time Period** to include the BegBalance time period in this scenario for currency conversion.
11. **Optional**: Select **Enabled for Process Management** to include this scenario in approvals.
12. Click **Save**.

Specifying Versions

Use versions to group data used by an application.

Task	Topic
Create a version.	See Creating Versions .
Edit a version.	See Editing a Version .
Delete a version.	See Deleting Versions .

About Versions

Use the Scenario and Version dimensions to create plans to be reviewed and approved. Each scenario/version combination contains data for accounts and other dimensions of each entity. After users enter data for an entity for a scenario and version, they can submit or promote the data for the entity to other users for review and approval. Use versions to:

- Allow multiple iterations of a plan
- Model possible outcomes based on different assumptions
- Manage dissemination of plan data
- Facilitate target settings

Target and Bottom Up Versions

You can create target and bottom up versions. With bottom up versions, you enter data into bottom level members; parent level members are display-only and do not permit data entry. Parent member values are aggregated from bottom level members.

For target versions, you can enter data for members at any level in the hierarchy. You can use business rules to distribute values from parent members to their descendants. Use target versions to set high-level targets for your plan. Planners working with bottom up versions can reference these targets when they enter plan data.

Target versions use top-down budgeting. Manage Approvals Tasks are not allowed, and children of target members must be blank (for example, #MISSING) to enable data input at the top level. Target members must be set to Store (Dynamic Calc overrides data input with sum of children).

Creating Versions

To create versions:

1. Select **Administration**, then **Manage**, then **Dimensions**.
2. Select **Version**.
3. Click **Add Child**.
4. For **Version**, enter the name of the version you are adding to the application.
5. For **Type**, select the type of version to display for the application:
 - **Standard Target** - Values are entered from the parent level down.
 - **Standard Bottom Up** - Values are entered at the lowest member level and aggregated upward.
6. **Optional:** For **Description**, enter a description.
7. **Optional:** For **Alias Table**, select an alias table to associate with the version, and enter a description.

8. **Optional:** Select **Enabled for Process Management** to include this version in approvals.
This option is not available for target versions.
9. **Optional:** Select **Enable for Sandboxes** so this member can be used for sandboxes.
10. Click **Save**.

Editing a Version

You can change the version name and access rights.

To modify versions:

1. Select **Administration**, then **Manage**, then **Dimensions**.
2. Select **Version**.
3. Select the version to edit.
4. Click **Edit**.
5. **Optional:** For **Version**, modify the name of the version.
6. **Optional:** For **Typelist**, modify the type of version to display for the application:
 - **Standard Target** - Values are entered from the parent level down.
 - **Standard Bottom Up** - Values are entered at the lowest member level and aggregated upward.
7. **Optional:** For **Description**, enter a description.
8. **Optional:** For **Alias Table**, select an alias table to associate with the version, and enter a description.
9. **Optional:** Select **Enabled for Process Management** to include this version in approvals.
This option is not available for target versions.
10. **Optional:** Select **Enable for Sandboxes** so this member can be used for sandboxes.
11. Click **Save**.
12. Update and validate business rules and reports.

Deleting Versions

You cannot delete versions that are used in planning units that are started or are assigned to axes on forms. You must remove references to versions from forms and assign another version to axes. At least one version must remain in the application.

To delete versions:

1. Select **Administration**, then **Manage**, then **Dimensions**.
2. Select **Version**.
3. Select the versions to delete.
4. Click **Delete**.
5. Click **OK**.
If planning units are started, you cannot delete versions associated with them.
6. Update and validate business rules and reports.

Displaying Versions



To display versions:

1. Select **Administration**, then **Manage**, then **Dimensions**.
2. Select **Version**.
3. For **Display**, select the version types to display.
 - **Standard Target**: Values are entered from the parent level down.
 - **Standard Bottom Up**: Values are entered at the lowest member level and aggregated up.

Sorting Version and Scenario Members

You can sort version and scenario members in ascending or descending order. Sorting members affects the outline itself.

To sort version and scenario members:



1. On **Dimensions**, select the Scenario or Version dimension.
2. For **Sort**:
 - To sort by ascending order, click .
 - To sort by descending order, click .
3. Click **OK**.

The next time you create or refresh the database, the outline is generated with members placed in the same order as on the Dimensions tab.

Moving Scenario and Version Members in the Dimension Hierarchy

You can change the order of scenario and version members in the dimension hierarchy.

To move a member's position in the dimension hierarchy:

1. On **Dimensions**, select the scenario or version to move.
2. Perform one action:
 - To move the member up, click .
 - To move the member down, click .
3. Click **OK**.

The next time you create or refresh the database, the outline is generated with members placed in the same order as displayed on the Dimensions tab.

Adding an Aggregate Storage Outline to a Planning Application

Oracle Hyperion Planning administrators can add an aggregate storage outline using Planning Application Administration.

About Aggregate Storage

Aggregate storage is the database storage model that supports large-scale, sparsely distributed data that is categorized into many, potentially large dimensions. Selected data values are aggregated and stored, typically with improvements in aggregation time. Aggregate storage is an alternative to block storage (dense-sparse configuration).

A key difference between aggregate storage and block storage is that Oracle Hyperion Planning requires a separate application for each aggregate storage database. Contrast that with a block storage outline application, which can have multiple databases in each application.

For detailed information about aggregate storage, see "Managing Aggregate Storage" in the *Oracle Essbase Administration Services Online Help*.

Aggregate Storage Outline Plan Type Characteristics

- Oracle Hyperion Planning does not generate XREFs on aggregate storage databases. XREFs can only be generated on block storage databases.
- Because Planning does not require all base dimensions on an aggregate storage database, approvals may not apply to the aggregate storage database if an approvals dimension is missing. If this is the case, normal security would apply.
- Dynamic time series members are not applicable for the Period dimension in an aggregate storage application.
- Creating and refreshing security filters are not applicable for aggregate storage databases.
- Use of the aggregate storage feature within Planning requires the customer to have the appropriate license for its use.

Process for Adding an Aggregate Storage Database to a Planning Application

To add an aggregate storage database to a Oracle Hyperion Planning Application:

1. Create an aggregate storage plan type. Perform one task:
 - Create an aggregate storage plan type during application creation. See [Creating Applications](#).
 - Add a new plan type using the plan type editor.
2. Add dimensions to the aggregate storage plan type. See [Working with Dimensions](#).

Note:

If a Currency, Years, Scenario, or Version dimension is valid for an aggregate storage plan type, the dimension members are also valid for an aggregate storage plan type.

3. Add dimension members. See [Working with Members](#).
4. Refresh the outline for the Planning application. See [Creating and Refreshing Application Databases](#).

5. Create a form using the dimensions associated with the aggregate storage database. See [Managing Forms](#).

Adding Plan Types

You can add a plan type by selecting **Administration**, then **Manage**, and then **Plan Types**.

Table 12-13 Number of Plan Types Allowed

Application	Generic Block Storage Plan Types	Module Block Storage Plan Types	Aggregate Storage Plan Types ¹	Total Plan Types
Core Oracle Hyperion Planning	3	NA	4	7
Oracle Hyperion Public Sector Planning and Budgeting	3	1	5	9

¹ One aggregate storage plan type for each block storage plan type, plus one consolidating aggregate storage plan type

After a plan type is added, it behaves like any other Planning plan type. If the plan type maps to an aggregate storage database, then aggregate storage limitations apply.

To add a plan type using the plan type editor:

1. From Planning, select **Administration**, then **Manage**, and then **Plan Types**.
2. Click **Add Plan Type**, and then complete the plan type details.

Note:

For an aggregate storage plan type, you must specify an application name to contain the database since an aggregate storage database must reside in its own application. Administrators creating aggregate storage plan types must ensure that all aggregate storage applications are unique across the enterprise.

3. Click **Save**.

Setting up Dynamic Time Series Members

You can use Dynamic Time Series (DTS) members to create reports that show period-to-date data, such as quarter-to-date expenses. DTS members are created automatically during application creation, and can be used with members of the Period dimension. To set up DTS, you enable a predefined DTS member and associate it with a generation number (and, optionally, an alias table and alias name). For example, to calculate quarter-to-date values, you can enable the Q-T-D member and associate it with generation number 2. You can then use the Q-T-D DTS member to calculate monthly values up to the current month in the quarter.

Note:

DTS is not supported for the Period dimension in an aggregate storage application.

Oracle Hyperion Planning provides eight predefined DTS members:

- H-T-D: History-to-date
- Y-T-D: Year-to-date
- S-T-D: Season-to-date
- P-T-D: Period-to-date
- Q-T-D: Quarter-to-date
- M-T-D: Month-to-date
- W-T-D: Week-to-date
- D-T-D: Day-to-date

 **Caution:**

Oracle recommends that you perform a backup before using the DTS feature. See [Backing Up Applications and Application Databases](#). If you are using the Y-T-D or P-T-D member, you must rename the Years or Period dimension so it does not conflict with the reserved Dynamic Time Series generation names, Years and Period. Before using Y-T-D, rename the Years dimension; before using P-T-D, rename Period. After doing so, you must update all application artifacts affected by these changes, such as member formulas and business rules, and any reports that reference the dimension by name.

The DTS members provide up to eight levels of period-to-date reporting. Your data and database outline determine which members you can use. For example, if the database contains hourly, daily, weekly, monthly, quarterly, and yearly data, you can report day-to date (D-T-D), week-to-date (W-T-D), month-to-date (M-T-D), quarter-to-date (Q-T-D), and year-to-date (Y-T-D) information. If the database contains monthly data for the past 5 years, you can report year-to-date (Y-T-D) and history-to-date (H-T-D) information, up to a specific year. If the database tracks data for seasonal time periods, you can report period-to-date (P-T-D) or season-to-date (S-T-D) information.

Oracle recommends that you avoid assigning time balance properties (such as First and Average) to members set for dynamic calculations if you plan to use the members in Dynamic Time Series calculations. Doing so may retrieve incorrect values for parent members in your accounts dimension.

For detailed information, see the *Oracle Essbase Database Administrator's Guide*.

To set up Dynamic Time Series members:

1. Select **Administration**, then **Manage**, then **Dimensions**.
2. Select the Period dimension, and click **DTS**.
3. Select **Enabled** for the DTS series to use: **H-T-D**, **Y-T-D**, **S-T-D**, **P-T-D**, **Q-T-D**, **M-T-D**, **W-T-D**, or **D-T-D**.
4. Select a generation.

The number of generations displayed depends on the number of generations in the time dimension. You cannot associate DTS members with the highest generation (the dimension root).


 **Note:**

Oracle Essbase considers the Period dimension in Planning as Generation 1, so take that into account when you set up Dynamic Time Series members.

5. **Optional:** Select an alias table and type an alias name. (If necessary, resize the window to view the fields.)
See [Working with Alias Tables](#).
6. Click **Save**.

Additional Supported Planning Application Features

Additional Oracle Hyperion Planning application features are supported for Planning dimensions. You can add children and siblings to Scenario, Version, and Period dimensions, and you can use the cut, paste, expand, and collapse features to work with their dimension hierarchies (see [Working with Dimension Hierarchies](#)). You can also use shared members for these dimensions, and can set two pass calculations at the root level for all dimensions. For example, you can:

Feature	More Information
In Scenario and Version dimensions, create hierarchies and use shared members. If you assign children to bottom-up versions, these versions display as read-only parents on forms.	See Setting Up Scenarios and Specifying Versions .
In the Period dimension, create alternate hierarchies and use shared descendants. Data Storage for all time periods can be set to any valid Data Storage value. The Consolidation operator for all time periods, including BegBalance, can be set to any valid consolidation operator. For example, it can be set to + instead of ~ (ignore).	See Working with the Years Dimension , Editing the BegBalance Member , and Editing Exchange Rate Tables .
Turn on two pass calculation at the root level, for example, for Account.	See Adding or Editing Members .
<p> Caution:</p> <p>Two pass calculation is ignored on any non-Account member not set to Dynamic Calc. When using this setting, consider the impact on currency conversion scripts.</p>	
For attributes, create hierarchies and assign aliases.	See Working with Attributes .

 **Note:**

For a multicurrency application, the pre-built currency conversion calc scripts does not function properly if you change data storage to dynamic for any Scenario, Version, Period, or Years member, or for certain dimension roots (such as Entity, Version, Currency, and custom dimensions). When changing data storage, consider the impact on currency conversion scripts.

Considerations for Alternate Hierarchies in Period Dimensions

If you create an alternate hierarchy in the Period dimension, the alternate hierarchy must follow the YearTotal member in the outline.

Working with Applications that Use Planning Application Administration

You can create and update applications that use Oracle Hyperion Planning application administration if you are assigned the Oracle Hyperion Shared Services Dimension Editor and Planning Application Creator roles. For information, see the *Oracle Enterprise Performance Management System User Security Administration Guide*.

Refreshing Application Databases Using a Utility

The CubeRefresh utility creates or refreshes the database. Before it runs, it ensures that the application is not locked. While it runs, the application is locked when metadata is updated in the database. For example, if users are assigning access, this message displays: "Cannot process your request because the application is being refreshed." This utility must be run on the same machine as the Oracle Hyperion Planning server, not on a remote server.

 **Caution:**

Oracle recommends that you back up the application before creating or refreshing. See [Backing Up Applications and Application Databases](#). Following these steps affects data in the database. When you use the create or refresh options, data might be replaced or erased, and Planning plan types rebuilt. For important information, see [Considerations for Working with Essbase](#).

To refresh the application database using a utility:

1. Back up the application. See [Backing Up Applications and Application Databases](#).
2. Locate the `CubeRefresh` utility.

The utility is in the `planning1` directory. For the full path to `planning1`, see [About EPM Oracle Instance](#).

3. Launch `CubeRefresh` from the `planning1` directory by entering the command and its parameters in the command line:

Windows: `CubeRefresh.cmd [-f:passwordFile] /A:application_name /U:user_name [/C|R] /D [/F[S|V]][/RMIPORT:rmi_port] [/L] [/DEBUG]`

UNIX: `CubeRefresh.sh [-f:passwordFile] /A:application_name /U:user_name [/C|R] /D [/F[S|V]][/RMIPORT:rmi_port] [/L] [/DEBUG]`

- **Optional:** If an encrypted password file is set up, you can specify `[-f:passwordFile]` as the first parameter in the command line, where `passwordFile` is the full file path and name for the password file. See [Suppressing Password Prompts in Planning Utilities](#).
 - `application_name`: The name of the Planning application on which to run the create or refresh.
 - `user_name`: The administrative user who has rights to create or refresh the application.
 - `/C` or `/R`: The function to perform on the application:
 - `/C`: Create the database outline.
 - `/R`: Refresh the database outline.
 - `/D`: Specify the database during create or refresh.
 - `/F`: Use security filters for all users of the application. Optionally, use with `s`, `v`, or `sv`:
 - `/FS`: Generate shared member security filters.
 - `/FV`: Validate security filters, but do not save the filters.
 - `/FSV`: Validate the shared member security filters.
 - `/RMIPORT`: Specify an RMI port number different from the preset value of 11333.
 - `/-L`: Default option, to create or refresh by connecting to an application server on a local or remote machine. (You can use `/L` to create or refresh without connecting to an application server, for example, if the application server is not running.)
 - `/DEBUG`: Specify detailed error messages.
4. If prompted, enter your password.
 5. View the result of the application refresh, including errors. Completion status is shown on the console.

This example shows a command line used to refresh a database for all users logged on to the specified application. It uses the default RMI port and includes shared member security filters:

Windows: `CubeRefresh.cmd /A:app1 /U:admin /R /D /FS`

UNIX: `CubeRefresh.sh /A:app1 /U:admin /R /D /FS`

Refreshing Databases with Level 0 Members Set to Dynamic Calc

If a level 0 member in the database outline is set to Dynamic Calc or Dynamic Calc and Store, Oracle Hyperion Planning successfully creates or refreshes, even if the members are not associated with a member formula. Planning adds a placeholder formula for members that do not currently have a member formula.

Working With Essbase Partitions

If you use Oracle Essbase Partitioning, partitions can access data shared between databases on different applications or servers. For information on partitions, see the *Oracle Essbase Database Administrator's Guide*.

Before refreshing application databases in Oracle Hyperion Planning, remove the Essbase partition definitions. This prevents overwriting the Essbase partitions or corrupting the Essbase database. After refreshing, you can redefine the Essbase partitions.

If you use replicated partitions, you can create the partition definition at replication time, run the partition, and drop the partition definition. You need not remove the partition definition during refresh because it is created when the partition is run.

Sorting Members Using a Utility

You can use the Oracle Hyperion Planning Sort Member utility to sort dimension members. `SortMember.cmd` is similar to sorting on the Dimensions tab in Planning. You can sort Entity, Account, Scenario, Versions, and user-defined custom dimensions. You cannot sort Period, Years, or Currency dimension members. The utility is useful for sorting dimension members after loading members into Planning. The `SortMember.cmd` utility uses a command line interface. Only administrators can run it.

To launch the `SortMember.cmd` utility:

1. From the `planning1` directory on the server where Planning is installed, enter this syntax:

```
SortMember [-f:passwordFile] servername username application member children|
descendants ascend|descend
```

For the full path to the `planning1` directory, see [About EPM Oracle Instance](#).

Table 12-14 SortMember Utility Parameters

Parameter	Description
<code>[-f:passwordFile]</code>	Optional: If an encrypted password file is set up, use as the first parameter in the command line to read the password from the full file path and name specified in <code>passwordFile</code> . See Suppressing Password Prompts in Planning Utilities .
<code>servername</code>	The server name where the Planning application resides
<code>username</code>	The name of the Planning administrator
<code>application</code>	The name of the Planning application containing the dimension members to sort
<code>member</code>	The parent member whose children or descendants to sort
<code>children descendants</code>	Whether to sort by children or descendants: sorting by children affects only members in the level immediately below the specified member; sorting by descendants affects all descendants of the specified member
<code>ascend descend</code>	Whether to sort by ascending order or by descending order

For example:

```
SortMember localhost admin BUDGET account200 descendants ascend
```

2. If prompted, enter your password.

If the application server or RMI service is running when the `SortMember` utility is launched, you might see some `java.rmi` or "Port already in use" errors. They do not affect the functionality of this utility.

Deleting Shared Descendants Using a Utility

Use the `DeleteSharedDescendant` utility to delete shared dimension members that are descendants of a given member. You can delete shared Entity, Account, and user-defined dimension members. All shared descendant members are deleted, not just immediate children of the specified member.

Administrators run this utility using a command line interface. If the application server or Oracle RMI service is running when the utility is launched, you may see `java.rmi` or "Port already in use" errors. They do not affect the functionality of the utility.

To use the `DeleteSharedDescendants` utility:

1. Launch the `DeleteSharedDescendants.cmd` file from the `planning1` directory on the server where Oracle Hyperion Planning is installed, using this syntax:

```
DeleteSharedDescendants [-f:passwordFile] servername username application member
```

For the full path to the `planning1` directory, see [About EPM Oracle Instance](#).

Table 12-15 DeleteSharedDescendants Utility Parameters

Variable	Description
<code>[-f:passwordFile]</code>	Optional: If an encrypted password file is set up, use as the first parameter in the command line to read the password from the full file path and name specified in <code>passwordFile</code> . See Suppressing Password Prompts in Planning Utilities .
<code>servername</code>	The server name on which the Planning application resides.
<code>username</code>	The name of the Planning administrator
<code>application</code>	The name of the Planning application containing the shared dimension members to delete.
<code>member</code>	The member whose shared descendants to delete. If a member name contains a space, surround it by quotation marks (for example, "Member One"). If the shared member itself is specified in the command line, the message "No shared descendants of <code>member_name</code> were found" is displayed.

For example:

```
DeleteSharedDescendants localhost admin BUDGET account200
```

2. If prompted, enter your password.
3. To see the results of running the utility, check the log files that are generated in the `EPM_ORACLE_INSTANCE/diagnostics/logs/planning` directory:
 - `DeleteSharedDescendants.log`: contains status messages.
 - `DeleteSharedDescendantsExceptions.log`: contains error messages.

If you run this utility while the Manage Database task is open, no error message is displayed, but the member is not deleted. The LOG and CMD files in the *EPM_ORACLE_INSTANCE/*diagnostics/logs/planning directory show that one shared member was found, but 0 shared members were deleted.

Using Smart Lists, UDAs, and Member Formulas

For applications that use Oracle Hyperion Planning application administration, you can create and update Smart Lists, UDAs, and member formulas.

Working with Smart Lists

Administrators use Smart Lists to create custom drop-down lists that users access from form cells. When clicking in cells whose members are associated with a Smart List (as a member property), users select items from drop-down lists instead of entering data. Users cannot type in cells that contain Smart Lists. Smart Lists display in cells as down arrows that expand when users click into the cells.

Perform these tasks to create and administer Smart Lists:

- Define Smart Lists, described here.
- Associate Smart Lists with members.
- Select dimensions for which Smart Lists are displayed.
- Optionally:
 - Use Smart List values in member formulas.
 - Set how #MISSING cells associated with Smart Lists display in forms.
 - Synchronize Smart Lists in reporting applications

To create or work with Smart Lists:

1. Select **Administration**, then **Manage**, then **Smart Lists**.
2. Perform one action:
 - To create a Smart List, click **Create**, enter the name, and click **OK**.
 - To change a Smart List, select it and click **Edit**.
 - To delete Smart Lists, select them, click **Delete** and **OK**. Deleting Smart lists also deletes any associated mappings with dimension members and reporting applications.

Data cells can display only one Smart List. If multiple Smart Lists intersect at cells, set which one takes precedence.
 - **Optional:** Click **Synchronize** to synchronize Smart Lists between an application that uses Oracle Hyperion Planning application administration and a reporting application. See [Synchronizing Smart Lists in Reporting Applications](#).

Synchronizing Smart Lists in Reporting Applications

For applications that use Oracle Hyperion Planning application administration that map Smart Lists to dimensions in reporting applications, you can synchronize Smart Lists in the Planning application. This identifies dimensions in reporting applications to which Smart Lists are mapped, and adds level 0 members of the selected dimensions as new Smart List entries to the selected Smart Lists. See [Mapping a Planning Application to a Reporting Application](#).

To synchronize Smart Lists in reporting applications:

1. Refresh the application database. See [Creating and Refreshing Application Databases](#).
2. Refresh the reporting application mapping. See [Mapping an Application for Reporting](#).
3. Select **Administration**, then **Manage**, then **Smart Lists**.
4. Click **Synchronize**, then click **OK**.

During synchronization, values from reporting applications in all existing mappings are appended after the last Smart List item in the appropriate Smart list. If a Smart List is mapped to two dimensions, all members from the first mapping are inserted first, and then members from the second mapping are inserted. If a member already exists in a Smart List, it is not added again. Smart List members in the Planning Smart Lists are not deleted, even if the corresponding dimension members on the reporting application are deleted.

 **Note:**

If Account is mapped as Smart List to Dimension, all of the level 0 members in the Account dimension are brought in as Smart List entries when the Smart List is synchronized. For example, Smart Lists may include entries such as HSP_Average and HSP_Ending. If this occurs, delete the extra entries from the Smart List. See [Adding or Changing Smart List Entries](#).

5. If Smart List items are mapped to more than one dimension, create a new Smart List with a new name, and then manually transfer related data.

 **Note:**

Smart List names cannot have spaces in them. If you are synchronizing Smart Lists in a reporting application, ensure that any new members do not have spaces in the name.

Adding or Changing Smart List Properties

Use the Edit Smart List Properties tab to set Smart List properties.

To set Smart List properties:

1. Select **Administration**, then **Manage**, then **Smart Lists**.
2. Select a Smart List and click **Edit**.
3. Define Smart List properties on **Properties**:

Table 12-16 Smart List Properties

Property	Description
Smart List	Enter a unique name containing only alphanumeric and underscore characters (for example: Position) and no special characters or spaces. Smart List names can be referenced in formula expressions.

Table 12-16 (Cont.) Smart List Properties

Property	Description
Label	Enter the text to display when the Smart List is selected. Spaces and special characters are allowed. Smart List labels can reference a resource, which can be translated into different languages. See About Customizing Text, Color, and Images .
Display Order	How Smart Lists are sorted in the drop-down list: by ID, Name, or Label
#MISSING Drop-Down Label	Enter a label (for example, "No Justification") to be displayed as an entry in the Smart List whose value is #MISSING. Notes: <ul style="list-style-type: none"> • It displays as the first selection in the Smart List drop-down, allowing #MISSING as a selection in the form. • When the cell is not in focus, this label displays only if Drop-Down Setting is selected in the next option. Otherwise, #MISSING or a blank cell is displayed, depending on the Display Missing Values As Blank selection for the form. • #MISSING labels determine only the display of cells with #MISSING data; #MISSING remains the stored value.
#MISSING Form Label	Determines how #MISSING values are represented in cells associated with Smart Lists. Options: <ul style="list-style-type: none"> • Drop-Down Setting: Displays the label set in #MISSING Drop-Down Label. • Form Setting: Displays #MISSING or leaves cells blank, depending on the Display Missing Values As Blank selection for the form. This selection determines what is displayed in the cell when it is not the focus. When the cell is in focus, the Smart List item that is selected from the drop-down is displayed.
Automatically Generate ID	Generate a numeric ID for each Smart List entry. If you do not select this option, you can customize Smart List ID values.

4. Click **Save**.

5. Select **Entries**.

Use the Entries tab to define selections on Smart Lists.

Adding or Changing Smart List Entries

Use the Edit /Add Smart Lists Entries tab to define the selections in the Smart List.

To define Smart List entries:

1. Select **Administration**, then **Manage**, then **Smart Lists**.

2. Select a Smart List and click **Edit**.

3. On **Entries**, define drop-down list items:
 - **For first items only:** enter information into the first row.
 - To add an item, click **Add** and enter the information.
 - To delete an item, select it and click **Delete**.
 - To edit an item, change the information in its row:

Table 12-17 Smart List Entries

Entry Property	Description
ID	Unique number that sets the order for the displayed entry. Customizable only if Automatically Generate ID is not selected on the Properties tab.
Name	Unique alphanumeric name containing alphanumeric and underscore characters (for example: Customer_Feedback) and no special characters or spaces
Label	Displayed text for the Smart List entry on the drop-down list (for example: Customer Feedback).

Items highlighted in red are duplicates.

4. Perform one action:
 - Click **Save**.
 - Select **Preview**.

Previewing Smart Lists

Preview the defined Smart List on the Preview tab. The tab shows the Smart List as displayed in a drop-down list or a table.

Displaying #MISSING with Smart Lists

Administrators set values displayed in Smart Lists and data cells, including the display when no data is in the cell. Cells can display no value, #MISSING, or (for cells associated with Smart Lists) a specified value.

Use these options to control the display of #MISSING when cells are not in focus:

Option	Guideline
Blank	When designing forms, select Display Missing Values as Blank . When setting Smart List properties, select Form Setting .
#MISSING	When designing forms, do not select Display Missing Values as Blank . When setting Smart List properties, select Form Setting .

Option	Guideline
A custom label, such as "No Change"	When setting Smart List properties, enter the custom label in the #MISSING Drop-Down Label field (for example, No Change). Select Drop-Down Setting .

Working with UDAs



You can use user-defined attributes (UDAs), descriptive words or phrases, within calc scripts, member formulas, and reports. UDAs return lists of members associated with the UDA. For example:

- You can use the `HSP_UDF` UDA to prevent a formula from being overwritten when the application is refreshed. You must log on to each database associated with the Planning application and create the member formula with a UDA. The syntax for this UDA is: (UDAs : `HSP_UDF`).
- If you use the `@XREF` function to look up a data value in another database to calculate a value from the current database, you can add the `HSP_NOLINK` UDA to members to prevent the `@XREF` function from being created on all plan types that are not the source plan type selected for that member.
- For a Product dimension with several product members, you can create a UDA called New Products and assign this UDA to the new products in the Product dimension hierarchy. Then you can base certain calculations on the designation New Products.
- For the budget review process, rather than creating a data validation rule for each owner in a product line (some product lines can have hundreds of owners), you can create a UDA containing the user names that apply to members using the planning unit hierarchy. Then in the data validation rule, you can enter a lookup function which will return the user names stored in the UDA for the current member. For example, create a UDA for each user in the promotional path and assign a prefix to the UDA name (for example, ProdMgr:Kim).

For more information on creating and using UDAs, see [Considerations for Working with Essbase](#) and the *Oracle Essbase Database Administrator's Guide*.

UDAs are specific to dimensions. For example, creating a UDA for an Account member makes it available for non-shared Account members. Deleting it removes it for all Account members. To make UDAs available for multiple dimensions, create the same UDA for multiple dimensions. For example, create a UDA named New for Account and Entity dimensions to make the UDA named New available for Account and Entity members.

To select UDAs for members:

1. Select **Administration**, then **Manage**, then **Dimensions**.
2. Select the dimension for whose members to associate the UDA.
3. From the dimension hierarchy, select a member and click **Edit**.
4. Select **UDA**.
5. **Optional:** To create a UDA, click **Create**.
6. Select UDAs for the member by moving them to **Selected UDA** and clicking **Save**:
 -  move selected UDAs
 -  remove selected UDAs

-  remove all UDAs

Creating UDAs

To create UDAs:

1. Navigate to the **UDA** tab.
2. On **UDA**, click **Create**.
3. Enter a name and click **Save**.

Note:

If creating a UDA for approvals, assign a prefix to the beginning of the name (for example, *ProdMgr:Name*). A prefix indicates that the UDA contains a user name and enables the data validation rule to look up the user. Use the same prefix for all approvals UDAs.

Changing UDAs

To change UDAs:

1. Navigate to the **UDA** tab.
2. On **UDA**, select a UDA and click **Edit**.
3. Change the name and click **Save**.

Deleting UDAs

Deleting a UDA removes it for the whole dimension.

To delete UDAs:

1. Navigate to the **UDA** tab.
2. Select the UDA and click **Delete**.

If you delete UDAs, you must update all member formulas, calc scripts, and reports that reference them.

Working with Member Formulas

You can define member formulas to combine operators, calculation functions, dimension and member names, and numeric constants to perform calculations on members. Member formulas can also include:

- Operator type, function, value, member name, UDA, and so on allowed in formulas.
See the *Oracle Essbase Database Administrator's Guide*.
- Predefined formula expressions, including Smart List values, that expand into a formula or value upon database refresh.

To define member formulas:

1. Select **Administration**, then **Manage**, then **Dimensions**.

2. Select the dimension for whose member to add or change a formula.
3. Select the member and click **Edit**.
4. Select the **Member Formula** tab.
5. Select options for the following fields:
 - **Plan Type**

 **Note:**

A formula entered for the default plan type will be applied to all plan types unless it is overridden by a different formula entered for a specific plan type.

- **Data Storage**—Select a data storage option. The default is **Store**.

 **Note:**

The plan type specific data storage field will not display the **Shared** or **Label Only** options. This is because a member cannot be set to Shared or Label Only in one plan type and not another.

- **Solve Order**—For aggregate storage plan types only, solve order specifies the order in which formulas are evaluated. Enter a whole number between 0 and 100000 (or use arrows to increase or decrease the number). The formulas for members that have a specified solve order are calculated in order from the lowest solve order to the highest. The default is 0.
6. In the text box, define formulas for the member.

See the *Oracle Essbase Database Administrator's Guide* for syntax, rules, and usage on Oracle Essbase formulas.
 7. **Optional:** To check the validity of the member formula, click **Validate**.
 8. Click **Save**.

Before you click **Save**, clicking **Reset** restores the previous member formula information.

Viewing Details of Formula Validation

To view details of the member formula validation:

1. On **Member Formula**, click **Validate**.
2. If the member formula is not valid, click **Show Details**.

If the member formula is valid, **Show Details** is not selectable.
3. Click **Save**.

Working with Formula Expressions

Oracle Hyperion Planning member formulas support Oracle Essbase native formulas and Planning formula expressions that are evaluated and expanded into Essbase code blocks when the database is refreshed. In these expressions, you can address Smart Lists by name, which Planning replaces with their numerical values in calculations.

In the text box on the **Member Formula** tab, you can include predefined formula expressions in member formulas, and test them with the **Validate** button. You can also load them.

You can update the dimension outline without updating the business rules and calc scripts that depend on the outline. Calculations become more independent of specifics in the outline. You can use Smart Lists as objects in calculations. Performance is not decreased when you use formula expressions because they are run only when you refresh the database.

To use a formula expression in a member formula:

1. Select **Administration**, then **Manage**, then **Dimensions**.
2. Select the dimension for whose member to add or change a formula.
3. Select the member and click **Edit**.
4. Select **Member Formula**.
5. Select options for the following fields:
 - **Plan Type**

 **Note:**

A formula entered for the default plan type will be applied to all plan types unless it is overridden by a different formula entered for a specific plan type.

- **Data Storage**—Select a data storage option. The default is **Store**.

 **Note:**

The plan type specific data storage field will not display the **Shared** or **Label Only** options. This is because a member cannot be set to Shared or Label Only in one plan type and not another.

- **Solve Order**—For aggregate storage plan types only, solve order specifies the order in which formulas are evaluated. Enter a whole number between 0 and 100000 (or use arrows to increase or decrease the number). The formulas for members that have a specified solve order are calculated in order from the lowest solve order to the highest. The default is 0.
6. In the text box, define formulas for the member.

You can include Planning formula expressions and Essbase native formulas in the member formula. See the *Oracle Essbase Database Administrator's Guide* for syntax, rules, and usage on Essbase native formulas.

Planning provides predefined formula expressions that you can use in member formulas. You cannot edit or create your own formula expressions.
 7. **Optional:** To check the validity of the member formula, click **Validate**.
 8. **Optional:** If there are errors in the member formula, click **Show Details** to view a description.
 9. **Optional:** Click **Reset** to restore the previous member formula if you do not want to save the changes you made to the member formula.
 10. Click **Save**.

Prerequisites

Before using formula expressions in member formulas, you should understand Oracle Essbase formulas and calculation and the application outline. See the *Oracle Essbase Database Administrator's Guide*.

Syntax

Member formula expressions support functions and variables. Follow these syntax rules for functions and variables when you create formula expressions:

- Enclose variables or properties with square brackets, []. If you omit square brackets, the variable is treated as a native variable.
- Enclose member names with quotation marks.
- Characters in variables are case-insensitive, but cannot have extra spaces or characters such as underscore (_).
- You can include subcalls to other functions within an expression.
- Do not enter text where a number is required.
- The order of the outline is important in a member formula. For example, do not reference a value that has not been calculated yet.

Including Smart List Values as Variables

You can include a Smart List as a variable in a formula expression, such as the formula expression, "Status"= [Status.Departed]

"Status" is the member name, Status is the Smart List name, and Departed is a Smart List entry. If the Smart List ID for Departed is 2, Status.Departed is replaced with a 2 in the member formula (Oracle Hyperion Planning treats Smart Lists as numbers). If the Smart List ID for Departed is 2, 2 is put in the calculation and 2 is stored in the database.

Write Smart Lists in this format: [SmartListName.SmartListEntry]

Formula Expressions

Oracle Hyperion Planning formula expressions can include these predefined variables and functions.

Table 12-18 Variables in Formula Expressions

Variable	Description
OpenInputValueBlock	Generates an IF statement if the Planning application is a multicurrency application, or an empty string if it is one currency application. Used with ClosedInputValueBlock.
ClosedInputValueBlock	Generates an End IF statement if the Planning application is a multicurrency application, or an empty string if it is one currency application. Used with OpenInputValueBlock.
NumberOfPeriodsInYear	Returns the number of time periods in the year
NumberOfYears	Returns the number of years in the application

Table 12-19 Functions in Formula Expressions

Function	Description
Dimension(dimTag)	Returns the name of a predefined dimension. The dimtags are: <ul style="list-style-type: none"> • DIM_NAME_PERIOD • DIM_NAME_YEAR] • DIM_NAME_ACCOUNT • DIM_NAME_ENTITY • DIM_NAME_SCENARIO • DIM_NAME_VERSION • DIM_NAME_CURRENCY
Period(periodName)	Returns the specified period. The periodName options are: <ul style="list-style-type: none"> • FIRST_QTR_PERIOD • SECOND_QTR_PERIOD • THIRD_QTR_PERIOD • FOURTH_QTR_PERIOD • FIRST_PERIOD • LAST_PERIOD
CrossRef(accountName)	Generates a cross reference for the account
CrossRef(accountName, prefix)	Generates a cross-reference for the account. The account name contains a prefix that you define. The default prefix is No, followed by a blank space and the account name, for example, No Salary.
getCalendarTPIndex()	Generates a member formula that returns an index for the time period; the index is based on the calendar year.
getFiscalTPIndex()	Generates a member formula that returns an index for the time period; the index is based on the fiscal year.
CYTD(memberName)	Generates a calendar year-to-date formula for the member
CYTD(memberName, calTpIndexName, fiscalTpIndexName)	Generates a calendar year-to-date formula for the member, and the time period index based on the calendar year and fiscal year. Use when members are renamed. The default member names are "Cal TP-Index" and "Fiscal TP-Index."

Understanding Common Errors

Follow the rules of syntax carefully. If formula expression syntax contains errors, error messages are returned after you validate the member formula. To get information about error messages, click **Show Details** on the **Member Formula** tab. The most common error message is "Failed to execute." This occurs when you use parameters in the expression incorrectly. These actions cause "Failed to execute" error messages:

- Entering the wrong number of parameters in the formula expression
- Misspelling member names, functions, or variable names
- Not surrounding member names with quotation marks
- Including numbers where strings are required

Customizing Planning Web Client

Customizing Reports

Oracle Hyperion Planning includes templates that control the layout and content of PDF reports of forms, form definitions, task lists, and planning units. You can use the templates as is. You can also customize the templates to add company logos, and special formatting for features such as shading, page size, orientation, font, font size, headers, percentage of page used for headers, number of data columns per page, and precision.

To customize reports, you must install and configure Microsoft Office Word 2000 or later and Oracle Business Intelligence Publisher Desktop. You can then use Word's BI Publisher menu to update information in the report using a sample .XML file. You can also use Word features to customize formatting. You make the template available by saving the .RTF file with the appropriate name, and placing it in the classpath or in the `HspJS.jar` file.

You can customize four report types, using the corresponding sample and template files. For information on creating the reports, see the related topics.

Report Type	Sample File Name	Template Name	Related Topics
Form	PlanningFormSample.xml	PlanningFormTemplate.rtf	See Creating Simple Forms .
Form Definition	PlanningFormDefSample.xml	PlanningFormDefTemplate.rtf	See Printing Form Definitions .
Task List	PlanningTaskListSample.xml	PlanningTaskListTemplate.rtf	See the <i>Oracle Hyperion Planning User's Guide</i> .
Planning Unit Annotations	PlanningPUAnnotationSample.xml	PlanningPUAnnotationTemplate.rtf	See Printing Planning Unit Annotations .

This topic gives general customization instructions. For detailed procedures, see the documentation installed with Word and BI Publisher. The BI Publisher install guide and user's guide are also available here:

http://download.oracle.com/docs/cd/E10091_01/welcome.html

To install BI Publisher Desktop:

1. Download the most recent version of BI Publisher Desktop from this location:

<http://www.oracle.com/technetwork/middleware/bi-publisher/overview/index.html>

To customize reports, you only need to install BI Publisher Desktop. BI Publisher is not required.

2. Save the zip file to your drive, and extract it with Use Folder Names selected.
3. Close all Microsoft Office applications.
4. Navigate to the directory where you extracted the zip file, and double-click the `setup.exe` application.

5. Follow the instructions in the BI Publisher installation wizard to perform a basic installation, accepting the default settings.

To customize reports:

1. Open the Planning HspJS.jar file, installed by default in products/planning/lib, and extract the sample file and corresponding template file.

For example, to customize the task list report, extract these files:

PlanningTaskListSample.xml and PlanningTaskListTemplate.rtf.

2. Save the sample and template files to a location where you can open them later.
3. In Microsoft Word, open the .RTF template file for the report to customize.

For example, for task list reports, open the PlanningTaskListTemplate.rtf template file.

4. From Microsoft Word's **Oracle BI Publisher** menu, select **Data**, then **Load Sample XML Data**, and open the sample file for the report.

For example, for task list reports, open the PlanningTaskListSample.xml sample file.

If Word does not display the BI Publisher menu, select the **Template Builder** toolbar. (For example, select **View**, then **Toolbars**, and then **Template Builder**. See the *Oracle Business Intelligence Publisher User's Guide*.)

5. Use the Word menus to customize the template.

For example, you can insert graphics and update fonts. For assistance with these tasks, see the Word documentation.

6. **Optional:** To update fields in the report, use Word's BI Publisher menus to add fields from the sample file.

For example, select **Oracle BI Publisher**, then **Insert**, and then **Field**, click a field in the **Field** dialog box, and drag the field into the template.

For assistance with these tasks, see the BI Publisher documentation. (In Word, select **Oracle BI Publisher**, then **Help**. In BI Publisher, click Help to view online help.)

7. When you finish customizing, select **Oracle BI Publisher**, then **Preview Template**, and select a format for previewing your changes. You can preview in any format. Click the close box to close the preview file.

8. In the template file, select **Save As**, and save the template as an .RTF file with the appropriate file name from the table.

For example, if you are customizing task list reports, save the file as

PlanningTaskListTemplate_Custom.rtf.

Type of Template	File Name
All forms	PlanningFormTemplate_Custom.rtf
Specific form	PlanningFormTemplate_Data_Form_Name For example, to apply the template to a form called Income, save the template as PlanningFormTemplate_Income.rtf.
Form definition	PlanningFormDefTemplate_Custom.rtf
Task list status reports	PlanningTaskListTemplate_Custom.rtf
Planning unit annotation reports	PlanningPUAnnotationTemplate_Custom.rtf

You must save the template file in a location on the classpath so it is available to the Web application server. Insert the template file in the `HttpJS.jar` file at the root level (the same level as the template files).

- Restart the application server to make the custom template available when users create PDF reports in Planning.

Customizing the Planning Tools Page

Administrators can customize the Oracle Hyperion Planning Tools page by adding links to commonly used tools for analyzing, tracking, and reporting on planning data. Planning users can then open Planning links from the Tools menu and click links to open pages in secondary windows in their browser. See [Specifying Custom Tools](#).

Customizing Cascading Style Sheets

In Oracle Hyperion Planning, cascading style sheets help define user interface (UI) themes, which control the appearance of Planning. You can customize a Planning cascading style sheet to meet your needs. Some common style sheet customizations include:

- Changing font colors to add emphasis or create a color-coding scheme
- Adjusting the background color of certain UI elements
- Inserting your organization's logo

Before customizing cascading style sheets, ensure that you have basic knowledge of cascading style sheets and are familiar with style sheet editor tools. Some settings are necessary for displaying portions of the product.

Changes to cascading style sheets are made on the Web application server, so they affect all connected users. Style sheets are not located in a default directory, rather the Web application server extracts the files to a temporary location at runtime. For information on where to find these files on the Web application server, see [Location of Cascading Style Sheet Files](#).

Planning primarily uses two cascading style sheets: a global sheet that defines Web UI elements common to multiple Oracle products, and one specific to Planning. The global cascading style sheet is called `global.css`. This table lists the major sections in the `global.css` file, and shows the sections that Oracle strongly recommends against customizing.

Table 13-1 Major Sections in the `global.css` File

Customizable Sections	Sections Not Recommended for Customizing
Non-Navigation Styles	Tadpole Menubar Tadpole MenuItem Menu Styles (order is important)
Tadpole Masthead	Tadpole Minimized Masthead
Tadpole Content Area	Tadpole Toolbar
Tadpole Logon Styles	Tadpole Viewpane
Tabs	Tadpole Tree
Tab Anchor Tags	
Tab Body	
TaskTabs	
Task Tab Body	

Table 13-1 (Cont.) Major Sections in the global.css File

Customizable Sections	Sections Not Recommended for Customizing
Groupbox	
Groupbox Anchor Tabs	
Buttons	
When Drawing Buttons in HTML	
When Using Button Tag	
When Using Input Tag	

To customize a cascading style sheet:

1. Extract the `HyperionPlanning.ear` file and then the `HyperionPlanning.war` file to a temporary location.
See [Location of Cascading Style Sheet Files](#).
2. Locate the `HspCustom.css` file (under the `custom` directory in the temporary location where the `HyperionPlanning.war` was extracted).
3. Customize `HspCustom.css` and save it to `HyperionPlanning.ear`.
4. You must redeploy for the changes to be picked up.

All Planning Web pages refer to the `HspCustom.css` file, and settings made here override those in `global.css` and `planning.css`.

To customize styles in forms to reflect members' hierarchical positions within the row or column axis, see [Customizing the Style of Row and Column Members in Forms](#).

Location of Cascading Style Sheet Files

The cascading style sheets for Oracle Hyperion Planning are extracted by WebLogic to a temporary location at runtime. Users wanting to make changes to these files must extract the `HyperionPlanning.ear` file and then the `HyperionPlanning.war` file to a temporary location before making any updates. Any changes made to the style sheets must be re-added to the `HyperionPlanning.war` file (which is part of `HyperionPlanning.ear`), and then `HyperionPlanning.ear` must be redeployed.

Modifying Cascading Style Sheets

When working with Oracle cascading style sheets:

- Before making changes, copy the original version and make it easily accessible.
- Styles common across multiple products are in `global.css`.
- Styles for products requiring unique styles are in `productname.css`.
- Some user interface controls are components of third-party technologies. Changing styles of third-party controls requires changing all associated files.
- Avoid making changes to styles that affect layout, including Padding, Borders, Text and Vertical Align, Position, and White-space.
- Changing `color` affects font color. Changing `background` affects the color of elements such as buttons.

- To change hyperlink text color, use the hyperlink style (`a:link`, `a:hover`, and `a:visited`).
- Toolbar button changes may require changes to all buttons in the product.
- Toolbar buttons rely on a dark-blue background to display properly.

Customization Examples

This section provides topics on common customizations.

Changing the Color of the Content Area Background

You can change the background color of the content area on the right side of the page by editing the Tadpole Logon Styles section of the `global.css` file. This table shows how to edit files to change the default color to white.

Table 13-2 Changing the Color of the Content Area Background

Default Content Area Background	White Content Area Background
<pre>.content table.content { background: #e5eaeef; }</pre>	<pre>.content table.content { background: #ffffff; }</pre>

Changing the Color of Hyperlinks and Visited Hyperlinks

You can change hyperlink color by editing the Non-Navigation Styles section of the `global.css` file. This table shows how to edit the file to change the default color to red.

Table 13-3 Changing the Color of Hyperlinks and Visited Hyperlinks

Black Hyperlinks and Visited Hyperlinks	Red Hyperlinks and Visited Hyperlinks
<pre>a:link, a:visited { color: #000000; }</pre>	<pre>.content table.content { background: #ff0000; }</pre>

Customizing the Style of Row and Column Members in Forms

You can customize Oracle Hyperion Planning cascading style sheets to display forms with different styles for the levels of members in rows and columns, depending on their hierarchical position in the form.

Note:

- In ADF mode, changes to `planning.css` have no impact.
- Customized styles are applied regardless of the dimension displayed in forms.
- Text alignment in rows is not supported.
- Styles are based on members' displayed hierarchical level, not necessarily Oracle Essbase relationships.
- Setting large font sizes on a row impacts the row's alignment.
- Customized styles are not reflected when forms are printed to PDF files.

To customize the style of row and column members:

1. Modify the `planning.css` file using the instructions in [Customizing Cascading Style Sheets](#).
2. In the `planning.css` file, customize these header tags:

Table 13-4 Header Tags in Planning.css File

Rows	Columns
<code>rowHeader_0</code>	<code>columnHeader_0</code>
<code>rowHeader_1</code>	<code>columnHeader_1</code>
<code>rowHeader_2</code>	<code>columnHeader_2</code>
<code>rowHeader_3</code>	<code>columnHeader_3</code>
<code>rowHeader_4</code>	<code>columnHeader_4</code>

The tags `rowHeader_0` and `columnHeader_0` affect the lowest-level class. The tags `rowHeader_4` and `columnHeader_4` affect the highest-level class. Members above level 4 display with the level 4 style.

Customizing Skins to Add Components

To customize an existing skin to add components:

1. Create a file called `trinidad-skins.xml`, and insert this code:

```
<skins xmlns="http://myfaces.apache.org/trinidad/skin"><skin><id>HspCustom.desktop/<id><family>HspCustom</family><extends>blafplus-rich.desktop</extends><render-kit-id>org.apache.myfaces.trinidad.desktop</render-kit-id><style-sheet-name>custom/HspCustom.css</style-sheet-name></skin></skins>
```
2. Specify the following in the code:
 - Family—The name of the new skin
 - Style-sheet-name—The custom css file that contains the style selectors
 - Extends— The skin being modified
3. Override the desired style selectors.
4. Perform the tasks in [Registering Customized Skins](#).

Registering Customized Skins

To register modified skins in applications:

1. Add the updated skin files to **HyperionPlanning.ear**.
2. Put **trinidad-skins.xml** in **HyperionPlanning.war\WEB-INF**.
3. Copy the new **HspCustom.css** to **HyperionPlanning.war\custom** to replace the existing css file.
4. Copy the updated .ear file to **EPM_ORACLE_INSTANCE_HOME\products\Planning\AppServer\InstallableApps**.
5. Redeploy to the application server.
6. Restart the application server and clear the browser cache.

7. In Planning, select **Administration**, then **Application**, and then **Properties**.
8. Create a new application property called **SKIN_FAMILY**.
9. Set the property value to reference the custom skins, such as HspCustom.
10. Log out, and then log back in.

About Customizing ADF Components

Components Skin style selectors in ADF enable you to customize the appearance of UI components. Style sheet rules include a style selector, which identifies an element, and a set of style properties, that specify the appearance of the components. ADF Faces components include these categories of skin style selectors:

- Global selectors: determine the style properties for multiple ADF Faces components
- Component selectors: Component-specific selectors are selectors that can apply a skin to a particular ADF Faces component

For more information about style selectors, see:

- http://docs.oracle.com/cd/E23943_01/apirefs.1111/e25378/toc.htm
- <http://www.oracle.com/technetwork/developer-tools/jdev/index-092146.html>
- http://docs.oracle.com/cd/E16764_01/web.1111/b31973/af_skin.htm

About Customizing Text, Color, and Images

You can customize text, colors, and images in the Oracle Hyperion Planning Web interface, and you can localize text.

- [Customizing Text, Color, and Images](#)
- [Customizing Text in Non-Latin Languages](#)
- [Restricted Characters in Customized Messages](#)

Customizing Text, Color, and Images

You can customize text, color, and images for these items in the Oracle Hyperion Planning Web interface:

- Labels and messages, using `HspCustomMsgs_en.template`
- Customizable colors and images, using `HspCustomImgs_en.template`

Whenever you add labels, you must add them to the `HspCustomMsgs` resource file. For example, when you add Smart List entries or menu items, include labels in the resource file. When your application is localized, update the corresponding resource file. For example, to localize an application into three languages, add labels to all three localized `HspCustomMsg` files (each with their language code in the filename) and include the translated words. Note:

- You must avoid certain characters when customizing text or messages. See [Restricted Characters in Customized Messages](#).
- Some colors are named, and others are given as hex or RGB values.
- Image file names and the `Images` directory are case-sensitive on some Web application servers and operating systems.

- You must store images added to the Web interface using `HspCustomImgs_en.properties` in the `Images` directory.
- The silent deployment implemented by WebLogic 8.1 does not extract files from EAR and WAR archives. You must manually extract, modify, and archive the customization files.
- To customize other types of images, see [Customizing Cascading Style Sheets](#).
- The procedure describes customizing English messages in the `HspCustomMsgs_en` file. To localize the application, update the file for the appropriate language.

To customize text, color, and images:

1. After extracting the `HyperionPlanning.ear` file to a temporary location, locate these files: `HspCustomMsgs_en.template` and `HspCustomImgs_en.template`.

These template files are in the `custom` directory where the `HyperionPlanning.war` file was extracted.
2. Copy and paste `HspCustomMsgs_en.template` and `HspCustomImgs_en.template` to a temporary location.
3. Rename the files `HspCustomMsgs_en.properties` and `HspCustomImgs_en.properties`.
4. Update the content of `HspCustomMsgs_en.properties` and `HspCustomImgs_en.properties`.

Each line in `HspCustomMsgs_en.properties` represents a customizable text resource. Each line in `HspCustomImgs_en.properties` represents a customizable color or image resource.

The lines start with the name of the resource, an equal sign, and the customizable resource. You need not include punctuation at the end of lines. Carriage returns signal the end of the resource.

For example, you can change the Tools menu label from "Tools: Analyze and Report" to "Tools: Additional Resources" by changing this line in the `HspCustomMsgs_en.properties` file:

```
LABEL_HOME_TOOLS=Tools: Analyze and Report
```

to:

```
LABEL_HOME_TOOLS=Tools: Additional Resources
```

5. Save and add the updated files to `HyperionPlanning.ear`, copy the file to the following location, and then perform a redeploy for the changes to be picked up:

```
EPM_ORACLE_INSTANCE_HOME\products\Planning\AppServer\InstallableApps\Common\HyperionPlanning.ear\HyperionPlanning.war\WEB-INF\classes\
```

6. Stop and restart the Web application server.

For information on customizing error messages for business rule calculations in the `HspCustomMsgs` file, see [Customizing Error Messages](#).

Customizing Text in Non-Latin Languages

You can customize text in the Oracle Hyperion Planning Web interface for non-Latin languages by updating the `HspCustomMsgs_en.template` file. This example shows how to update the file for the Russian language.

To customize `HspCustomMsgs_en.template`:

1. Locate `HspCustomMsgs_en.template` in the `/custom` directory.

2. If you are customizing the labels for a specific language, such as Russian, rename the template file to use a `.source` extension, such as `HspCustomMsgs_ru.source`.
3. Review the source file for the language, such as `HspCustomMsgs_ru.source`, to determine which labels in the file need to be customized. Add the translation for the labels to the `.source` file, remove all other labels, and save the file. (You need only customize the labels in this file.)
4. Perform one action:
 - For languages such as Russian, Japanese, Korean, Turkish, Simplified Chinese, and Traditional Chinese, continue to step 5 to create the Java property file.
 - For Latin1 languages, such as Danish, German, Spanish, French, Italian, Brazilian Portuguese, and Swedish, create the property file manually by saving the `.source` file (such as `HspCustomMsgs_ru.source`) with a `.properties` extension, such as `HspCustomMsgs_ru.properties`.
5. Complete this step to use the `native2ascii` program to convert source strings to Unicode strings to create a custom property file for the language, such as `HspCustomMsgs_ru.properties`. To use this program, you must enter the encoding for the language, such as `Cp1251` for Russian. The usage:

```
native2ascii [-encoding language_encoding] [inputfile_name[outputfile_name]]
```

Table 13-5 Examples of Language Encoding Parameters

Language	Encoding Parameter
Russian	Cp1251
Turkish	Cp1254
Japanese	SJIS
Simplified Chinese	EUC_CN
Traditional Chinese	Big5
Korean	EUC_KR

- a. Ensure that the `native2ascii` program included with the Sun JDK is installed, and note the installation path to this program. Some application servers install this program for you. If it is not installed, you can install the JDK.
- b. Open a command prompt.
- c. Type the path to the `native2ascii` program, followed by the language encoding parameter and the names of the source and target files to be created. For example:


```
C:\j2sdk1.4.2_15\bin\native2ascii -encoding Cp1251 HspCustomMsgs_ru.source
HspCustomMsgs_ru.properties
```
6. Extract the `HyperionPlanning.ear` file to a temporary location.
7. Copy the `properties` file (such as `HspCustomMsgs_ru.properties`) to the `custom` directory where `HyperionPlanning.war` was extracted.
8. Re-create `HyperionPlanning.ear` and place the ear file in the following location:


```
EPM_ORACLE_INSTANCE_HOME\products\Planning\AppServer\InstallableApps\Common
```
9. Remove the extracted `\HyperionPlanning` from the temporary location.
10. Restart Planning.
11. Restart the Web application server.

Restricted Characters in Customized Messages

You can modify text strings to meet your business needs. Avoid using characters that can cause errors in custom messages, depending on how the text string is used and whether the string is generated by Java or JavaScript. (In most cases, text messages are generated by Java, and pop-up boxes and some buttons are generated by JavaScript.) For example, errors occur if you add this string inside a JavaScript call because JavaScript cannot parse double quotation marks:

```
someJavaScript("<%= HspMsgs.SOME_RESOURCE %>");
```

The best practice is to avoid using these characters in custom messages:

- Single quotation marks
- Double quotation marks
- Back slashes
- Forward slashes

Customizing Colors, States, and Actions for Process Status

These procedures require knowledge of how to maintain and manipulate your relational database:

- [Customizing Colors](#)
- [Customizing States](#)
- [Customizing Actions](#)

Customizing Colors

By default, planning unit status (for example, Under Review or First Pass) is displayed in black. You can customize each state to display in another color. Valid colors include all named colors that browsers recognize. See the Microsoft Web site for supported named colors.

The information for process status state colors is stored in the `HSP_PM_STATES` table in the `COLOR` column. You must run statements for your relational database to change the color value.

After you customize the colors, restart the Web application server.

Example: SQL Relational Database

Run this query to list all available process status states and their current color settings:

```
select  
* from hsp_pm_states
```

This returns the `state_id`, the name of the process status state, and the color. By default, the color value is `<NULL>`, which converts to black.

Note the `state_id` of the process status state you want to change, and then run this query:

```
UPDATE HSP_PM_STATES SET COLOR = 'new color' WHERE STATE_ID = state_id
```

Run this SQL statement to change the First Pass process status state color to green:

```
UPDATE HSP_PM_STATES SET COLOR='GREEN' WHERE STATE_ID=1
```




Note:

For Oracle relational databases, issue a `COMMIT;` command to commit the transaction.

Customizing States

You can customize these preset process status states:

- Not Started
- First Pass
- Under Review
- Frozen
- Distributed
- Signed Off
- Not Signed Off
- Approved

The process status state information is stored in the `HSP_PM_STATES` table in the `NAME` column. You must run statements for your relational database to change the state value.

Example: SQL Relational Database

Run this query to list all available process status states and their current names:

```
select * from hsp_pm_states
```

This returns the `state_id`, the name of the process status state, and the color.

Note the `state_id` of the process status state you want to change, and then run this query:

```
UPDATE HSP_PM_STATES SET NAME = 'NewName' WHERE STATE_ID = state_id
```

Run this SQL statement to change the Approved process status state name to Promoted:

```
UPDATE HSP_PM_STATES SET NAME='PROMOTED' WHERE STATE_ID=1
```



Note:

For Oracle relational databases, issue a `COMMIT;` command to commit the transaction.

Customizing Actions

You can customize these preset process status actions:

- Originate
- Start
- Promote

- Submit, Submit to Top
- Exclude
- Reject
- Approve
- Delegate
- Take Ownership
- Freeze, Unfreeze
- Distribute, Distribute Children, Distribute Owner
- Sign Off
- Reopen

The process status action information is stored in the `HSP_PM_ACTIONS` table in the `NAME` column. You must run statements for your relational database to change the action value.

Example: SQL Relational Database

Run this query to list all available process status actions and their current names:

```
select * from hsp_pm_actions
```

This returns the `action_id` and the name of the process status action.

Note the `state_id` of the process status action you want to change, and then run this query:

```
UPDATE HSP_PM_ACTIONS SET NAME = 'NewName' WHERE ACTION_ID = action_id
```

Run this SQL statement to change the Approve process status action name to Accept:

```
UPDATE HSP_PM_ACTIONS SET NAME='ACCEPT' WHERE ACTION_ID=1
```



Note:

For Oracle relational databases, issue a `COMMIT;` command to commit the transaction.

Creating Spreading Patterns

Administrators who understand SQL can create custom spreading patterns, accessible from the Grid Spread and Mass Allocate dialog boxes, by adding them to the `HSP_SPREAD_PATTERN` database table.

To add a custom spreading pattern:

1. Open the `HSP_SPREAD_PATTERN` database table.
2. Enter row values to name and represent how values are spread from a parent cell:

Table 13-6 HSP_SPREAD_PATTERN Table

Column	Description
NAME — VARCHAR (80) UNIQUE NOT NULL	Internal name (not displayed)
LABEL — VARCHAR (80) NOT NULL	Reference to a string ID in the resource file, which displays in the user interface. If a string resource is not created, the LABEL string identifier displays instead (for example "Label_Spread_445").
PATTERN VARCHAR (255) NOT NULL	Space delimited; one or more elements, up to 255 characters: <ul style="list-style-type: none"> • A number: For example, to specify that the parent value splits evenly across its children, specify 1. • !: Lock; do not change the value of a cell that has "!" as its pattern element. • P: Parent value being spread. A child receives the exact value the parent receives. • *: Repeat the element that follows. Use 1* to repeat 1 for every cell, to take up the rest of the space in the applied pattern. If no * exists as part of an element within a pattern, the entire pattern repeats itself instead of a specific element. A pattern cannot contain only the * character, and can have only one element with the * character. For example, 1 2* 3 is allowed, but 1 2* 3* is not.

3. Save and close the table.

The new patterns display as spreading options on the Grid Spread and Mass Allocate dialog boxes.

See also the *Oracle Hyperion Planning User's Guide* and [Examples of Spreading Patterns](#).

Examples of Spreading Patterns

Assume a form cell having three children cells is affected by a spread. Examples of how patterns are applied to the three child values:

Table 13-7 Examples of Applied Spreading Patterns

Pattern	Cell 1	Cell 2	Cell 3	Explanation
4 4 5	4	4	5	The new value is spread to level zero members using the 4 4 5 pattern. For example, if you set Q1 to 13, its value is spread as: Jan = 4 Feb = 4 Mar = 5

Table 13-7 (Cont.) Examples of Applied Spreading Patterns

Pattern	Cell 1	Cell 2	Cell 3	Explanation
1	1	1	1	1 is the same as 1*. This FLOW spread type shows how FLOW works if no values exist to make the spread proportional. The parent's value is divided by 3, the number of children cells, and spread equally to each of its three children.
P	P	P	P	Each child cell receives the parent's value. This is more a copy pattern than a spread.
!* P	!	!	P	Equivalent to a BALANCE type of spread.
P !*	P	!	!	Equivalent to a FIRST type of spread.
1 2 1	1	2	1	Similar to a bell-curve type of spread.
0 1 0	0	1	0	The beginning and ending children cells receive no spread values, and the middle child member receives the parent's value.

Creating Instructions for Planners

Using Oracle Hyperion Planning features, administrators can create customized instructions to guide planners in their work:

- Create explicit instructions in the Description text box when creating forms. For example, "Select this form to forecast revenue income." See [Editing Forms](#).
- Provide clear instructions for task lists and individual tasks, such as, "Use this task list to allocate general expenses" or "Select this task to allocate phone expenses."
- When setting options for forms, select the Enable Cell-level Document option (see [Setting Form Layout](#)), and then link cells to documents in Oracle Hyperion Enterprise Performance Management Workspace. These documents can be a Web site or any file type (for example, an .XLS or .PDF file). For example, you could associate a cell with a document that explains the organization's assumptions for January's telephone expenses. See the *Oracle Hyperion Planning User's Guide*.

- Create tasks of type:
 - URL—to open a URL. For example, link to the Accounting department's internal Web site that provides assumptions for the latest forecast.
 - Descriptive—to display instructions. For example, "If you travel during this period, leave a contact number with your Finance representative."
- Create descriptive text for runtime prompts that are specific and tell users what type of data is expected. For example: "Select the project," "Enter the expected number of customer visits per quarter," and "What percentage change in revenue do you expect next month?" See [Setting Business Rule Properties](#).
- Use broadcast messaging to send instructions to all planners who are currently logged on to an application. For example, broadcast a message, "Remember that all forecast revisions are due by the end of the week." See [Using Broadcast Messaging](#).

14

Troubleshooting

For detailed troubleshooting information, see the *Oracle Enterprise Performance Management System Installation and Configuration Troubleshooting Guide*.

Calculation Script is Too Long or Complex

Scenario:

Use the **Manage Currency Conversion** page to generate application-wide calculation scripts that you can launch from the Web to convert currencies. When generating calculation scripts, if the following error displays, the calculation script exceeds the 64K limit. Calculation script is too long or complex This can occur if an application has many periods (such as 500) included in a currency conversion calculation script.

Solution:

To resolve this issue:

1. Limit the calculation script to one scenario.
If the calculation script generates successfully, skip to the last step.
2. If the error still displays, limit the calculation script to one reporting currency.
If the calculation script generates successfully, try selecting two reporting currencies. If that works, continue adding reporting currencies until the error displays. Then skip to the last step. If the calculation script does not generate with one reporting currency, remove some currencies from the application.
3. Use the **Manage Currency Conversion** page to generate as many other, smaller currency conversion calculation scripts as necessary to include your scenarios, reporting currencies, and versions.

You can also give periods very short names.

Cannot Find Member

Scenario:

During database refresh, this error displays: "Cannot Find Member x."

Solution:

Perform a full database refresh (instead of an incremental refresh) from the **Refresh Database** page.

Cannot Process Request

Scenario:

This error displays to application users: "Cannot process your request because the application is being refreshed." While application databases are created or refreshed, aspects of the application are locked, and users cannot change:

- Dimensions or members
- Users
- Groups
- Security
- Aliases
- Alias tables
- Exchange rates
- Year
- Period

Solution:

Wait until the database is created or refreshed.

Conflicting Change by Users

Scenario:

When changing data, this error displays: "You are trying to change data that has been changed by a user on another server."

Solution:

The data is currently being changed by a user on another server. Wait a few seconds and try again. If the message continues to display, try closing the page and reopening it. If the error continues, restart the Web application server.

Copy Version Error

Scenario:

After using Copy Version with a large number of members selected, this error displays: An error occurred when running specified calc script, please check logs for details.

The log for the Web application server contains this message:

```
com.hyperion.planning.olap.HspOlapException: The length of the calc script exceeds the maximum allowed length.
```

Solution:

Copy Version uses a calculation script, and Oracle Essbase limits the size of calculation scripts to 64K. If the limit is reached while copying a large number of members, Copy Version can fail. To avoid this, select fewer members to copy with Copy Version. See the *Oracle Hyperion Planning User's Guide* and *Oracle Enterprise Performance Management System Installation and Configuration Troubleshooting Guide*.

To prevent a failed Copy Version, estimate how many members you can successfully copy. The script uses member names, so you must consider the number of members and the length of member names. If each member is an average of 10 characters, fewer than 6,400 members can be copied. The script also includes syntax characters, such as the calculation script command and a comma for each member name. Assuming a calculation script command length of 500 characters, you can use this formula:

$(\text{Number of members} * \text{Average length of member names}) + \text{Number of members} + 500 \leq 64,000$

Currency Conversion Calculation Script Failed

Scenario:

When verifying calculation scripts, this error displays if scenarios, versions, or currencies in the calculation script are Dynamic Calc members or all Account members are dynamic: "The FIX statement cannot contain a Dynamic Calc member."

Solution:

When selecting scenarios, versions, and currencies for currency conversion calculation scripts on the Manage Database page, do not select Dynamic Calc members. In addition, at least one account must be set to Store.

Form Error Message

Scenario:

The first person to use Oracle Hyperion Planning after an Oracle Essbase timeout may receive an error that the form failed to open.

Solution:

The user should click **Refresh** to restore the connection between Planning and Essbase.

Database Locked by Another Administrator

Scenario:

Occasionally, Oracle Hyperion Planning applications may become locked. This can be caused by events such as abnormally exiting the application.

Solution:

See [Unlocking Applications](#).

500 Error Message

Scenario:

You may receive a 500 ERROR message on a form when the number of members on the page (the cross-product combination of dimensions after security filters are applied) causes Java to run out of memory.

Solution:

Oracle recommends reducing the number of cross-product dimension combinations on the page as much as possible. By default, Java is allocated 128 MB of memory, but if your server has more memory available, you should increase the amount that Java can use. As a general rule, if the server is dedicated to Oracle Hyperion Planning, allocate 75% of the server's RAM to Java; if the server is not dedicated, allocate 25% to 50% of the RAM to Java. The minimum memory setting should be 128 MB.

For instructions, see the "Increasing the JVM Memory Allocation" section of the *Oracle Enterprise Performance Management System Installation and Configuration Guide*.

Implied Sharing Issues

Scenario:

For members that have an implied sharing relationship, if a parent and child are displayed on the same Oracle Hyperion Planning form, only values entered for the parent are retained. In the following examples, Planning creates an implied share relationship between Parent A and Child 1 because the values of the parent and child are always the same. These examples assume that all of the members are set to the Store data type.

Example 1:

```
Parent A
  Child 1 (+)
```

Example 2:

```
Parent A
  Child 1 (+)
  Child 2 (~)
  Child 3 (~)
```

Because most Planning applications are bottom-up applications, data is usually entered for the child because the parent is read only. The typical sequence of events:

1. The form displays the child, usually above the parent.
2. New data is entered for the child.
3. The form is saved. The save operation reads the form from left-to-right and top-to-bottom, so the child is saved first.
4. The save operation then takes the last occurrence of the value in the grid (the bottommost, rightmost value), which, because of the implied share, overwrites the value of the child. The data entered for the child is discarded.

Solution:

Depending on the requirements for your Planning forms, you can use these methods to avoid implied shares.

- For a parent and child on the same form: Add a dummy member as an aggregating child. The dummy member is included in the outline but is not used on forms. Implied sharing is disabled when the parent has only one aggregating child.
- For a Label Only parent: An implied share exists with the first child member regardless of how many aggregating children are present. To disable implied sharing in this situation, change the Label Only storage type or avoid including the parent and child on the same form.
- For a parent that can be set to Never Share: If necessary for your application, you can set the parent member to the Never Share storage setting. The Never Share parent functions similarly to a Store parent with multiple aggregating children. However, unlike a Store parent, a Never Share parent displays only the aggregated value of its children after an aggregation is run.

**Note:**

For parents with single children, using the default storage type of Store (keeping the implied share relationship) is usually advantageous, because doing so reduces the number of blocks that are created, the database size, and the calculation and aggregation times. Use Never Share only when necessary.

For detailed information on implied sharing, see the *Oracle Essbase Database Administrator's Guide*.

Manage Database Errors

Scenario:

Manage database errors can occur when the 8.3 naming convention is not used. When the application database is created or refreshed, the error may display as: "Manage Database Errors detected (Start 1060044): Could not open outline -2147218904."

Solution:

To resolve Manage Database errors, you may need to redirect the operating system environment variables. Oracle Essbase requires an 8.3 character-naming convention for the Temp and Tmp environment variables in the operating system. Use this procedure to check the naming convention and reassign these variables if needed.

File Naming Convention

Ensure that you are logged on to the Oracle Hyperion Planning server before making changes.

To check the environment variables for Windows 2000:

1. Close Planning.
2. Close Essbase.

3. On the Windows desktop, right-click **My Computer**.
4. Select **Properties**.
5. Select **Advanced**.
6. Select **Environment Variables**.
7. Change the settings for Temp and Tmp in the user variable to the 8.3 naming convention.
For example, C:\Temp
8. Open, and open the outline.
9. Close, and restart Planning.
10. Try creating or refreshing the database on the Manage Database page. If the error recurs, repeat the previous steps, but this time also create a folder called C:\tmp and set the System Environment variables **Temp** and **Tmp** to C:\tmp.

If the error persists, create a Temp folder under C:\temp. Repeat the previous steps, redirecting first the user variables then the system variables to C:\temp\temp. If the error continues, redirect the environment variables to C:\temp.

 **Note:**

The settings for Temp and Tmp in the user and system variables must be unique. Do not set the user and system variables to the same folder.

Maximum Number of Applications Already Connected or Database is Locked

Scenario:

You are using DB2 as your relational database, and you receive either of these messages when you try to create or refresh the database:

- ... maximum number of applications already connected
- Database is locked ...

By default, the `MAXAPPLS` parameter is set to 40.

Solution:

Increase the number of applications allowed by increasing the `MAXAPPLS` parameter.

To increase the `MAXAPPLS` parameter:

1. In **Control Center**, right-click the database and select **Configure Parameters**.

Alternately, you can set the parameter from the DB2 window.

2. Set the `MAXAPPLS` parameter, using this format:

```
db2 -v update db cfg for database name using MAXAPPLS n
db2 -v terminate
```

where *database name* is the name of the relational database and *n* is the number of applications that can be connected simultaneously. For example, to increase the number of maximum number of applications that can be connected simultaneously to 100 for a database named Business, specify:

```
db2 -v update db cfg for Business using MAXAPPLS 100
db2 -v terminate
```

Optimizing WebLogic Server Parameters

Scenario:

If the following conditions apply to your environment, then check the WebLogic Server settings described in the Solution.

- Your application's performance is slow.
- You have tried the solutions in [Optimizing Performance](#).
- Your application uses WebLogic Server.

Solution:

To tune WebLogic Server performance settings:

1. Open the WebLogic Server Administration Console.
2. Select the deployed Oracle Hyperion Planning application, the **Configuration** tab, the **Tuning** tab, and then set these values:
 - **Servlet Reload Check:** -1
This option sets the number of seconds that WebLogic Server waits to check whether a servlet was modified and needs to be reloaded. The value -1 means to never check.
 - **Resource Reload Check:** -1
This option sets the number of seconds that WebLogic Server waits to check whether a resource was modified and needs to be reloaded.
 - **JSP Page Check:** -1
This option sets the number of seconds that WebLogic Server waits to check whether JSP files have changed and needs to be recompiled.
3. Open the properties of the server, select the **Tuning** tab, and then select **Enable Native IO**.
Selecting this option ensures that WebLogic Server uses native performance packs. By default, when you are in Production mode, using native performance packs is enabled in `config.xml`.
4. To turn off HTTP access logging, open the server properties, select the **Logging** tab, the **HTTP** tab, and then clear **HTTP access log file enabled**.
By default, WebLogic Server enables HTTP access logging. Because the Web Server already creates an HTTP access log, this step turns it off.

▲ Caution:

HTTP access log files grow proportionally with the number of requests. If you do not configure log rotation properly, the access log file can grow very large and potentially fill up the disk, causing the application and the operating system to stop working. When this condition occurs, to remove the log files, you must stop and then restart the application server.

Optimizing Windows Network Parameters

Scenario

Network performance is slow on Windows systems.

Solution

Tune the Windows operating system to optimize network performance.

To optimize Windows performance:

1. Open the Windows registry and set the `TcpTimedWaitDelay` parameter to 30.

This step decreases the time that the operating system waits to reclaim a closed port from its default setting of 4 minutes (240 seconds). The parameter is at `HKLM\System\CurrentControlSet\Services\Tcpip\Parameters`

Parameter Name: `TcpTimedWaitDelay`

2. Set the `MaxUserPort` parameter to 65534.

This step increases the number of ports that can be opened on an application from its default of 5,000. The parameter is at

`HKLM\System\CurrentControlSet\Services\Tcpip\Parameters`

Parameter Name: `MaxUserPort`

Registering a Restored Application with Planning

Scenario:

You have restored a Oracle Hyperion Planning application, but it is not available for selecting in Oracle Hyperion Enterprise Performance Management Workspace.

Solution:

Register the application with Oracle Hyperion Shared Services from within Planning.

To register a restored Planning application with Shared Services from Planning:

1. In your browser, enter the URL for the EPM Workspace Log On page
2. In EPM Workspace, select **Navigate**, then **Applications**, then **Planning**. Select a Planning application. If prompted, enter your logon information.
3. Select **Administration**, then **Application**, then **Settings**.
4. From **Show**, select **Advanced Settings**, then click **Go**.

5. Click **Register Shared Services**.

Session Timeout

Scenario:

If a user quits a session by closing the browser instead of logging off, the session times out after a period of time, and the user name is removed from the Statistics page. You can change the number of minutes before the session times out by modifying timeout settings for your Web application server. Modify the `web.xml` file in `HyperionPlanning.ear` or `HyperionPlanning.war`.

Solution:

To change the session timeout setting:

1. Open the `web.xml` file in `HyperionPlanning.ear` or `HyperionPlanning.war`.
2. Modify the timeout setting and save your changes.

For example, change the number 60 in the `session-timeout` setting to the number of minutes to use.

To modify timeout settings, see the documentation for your Web application server.

Slow Performance When Opening Forms Using a Dial-Up Connection

Scenario:

Opening a form using a slow network connection (for example, with a modem) is slow.

Solution:

You can increase the network bandwidth when opening forms by modifying the `web.xml` file. This compresses by approximately 90% the data stream sent from the Oracle Hyperion Planning server to the client.

Note:

If you use a WebLogic (all supported versions) Web application server, complete the second procedure, specific to WebLogic. If you use another Web application server, complete the first procedure.

To improve performance for a Web application server other than WebLogic:

1. With a text editor, open the `web.xml` file in `HyperionPlanning.ear` or `HyperionPlanning.war`.
2. After the tag `</description>` and before the tag `<listener>`, insert these lines:

```
<filter>
```

```
<filter-name>HspCompressionFilter</filter-name> <filter-  
class>com.hyperion.planning.HspCompressionFilter</filter-class>  
  
<init-param>  
  <param-name>compressionThreshold</param-name>  
  <param-value>2048</param-value>  
</init-param>  
  
<init-param>  
  <param-name>debug</param-name>      <param-value>1</param-value>  
</init-param>  
</filter>  
  
<filter-mapping>  
  <filter-name>HspCompressionFilter</filter-name>  
  <url-pattern>/EnterData.jsp</url-pattern>  
</filter-mapping>
```

3. Save the web.xml file.

If you use WebLogic, you must manually modify the .ear file and redeploy it for the Web application server.

To improve performance with a WebLogic application server:

1. Unzip the HyperionPlanning.ear file to /ear, for example.
2. Unzip Hyperion.war under /ear to /war.
3. With a text editor, open /war/WEB-INF/web.xml and modify it using the instructions in step 2 in the preceding procedure.
4. Compress the content in /war to /ear/HyperionPlanning.war.
5. Compress the content in /ear into /ear/HyperionPlanning.ear.
6. Deploy the new HyperionPlanning.ear for the WebLogic Web application server.

Unable to Create Application Error

Scenario:

When you try to create an application in Oracle Hyperion Planning, this error might display: Unable to create application *application name*. Error number: -2147160060.

Solution:

There may be several causes. To resolve this, ensure that these conditions are met:

- Oracle Essbase must be running.
- Advanced User Rights must be configured on the local machine.
- The Administrator user must be configured for HsxServer and HspDataSource.
- If the Essbase supervisor user name matches the Planning administrator user name, the passwords must be identical.

- If you use local authentication, the machine name must be different from the user name.
- The Planning application name cannot match Essbase application names.
- If you are using DB2, a error message might state that the database is not configured correctly. This can result from various causes, such as incorrect configuration for auditing tables.

Unable to Refresh Application Error

Scenario:

When using a DB2 database and refreshing an application in Oracle Hyperion Planning, an error displays.

Solution:

DB2 could not refresh or correctly generate its transaction log because insufficient space was allocated for log files. (See the *Oracle Enterprise Performance Management System Installation and Configuration Troubleshooting Guide*.) To resolve this issue, use these procedures.

Allocate more space for the DB2 transaction log:

1. In DB2 Command Center, select **Tools**, then **Wizard**, and then **Configure Database Logging Wizard**.
2. Connect to the database and ensure that **Circular Logging** is selected.
3. On the next screen, increase the number of primary (for example, 20) and secondary log files (for example, 10).
4. Increase the size of each log file (for example, 2000).
5. Click **Next** to accept all defaults.
6. At **Summary**, enter the user name and password.
7. Click **Finish**.
8. Restart DB2.
9. Refresh the database.

Perform a database refresh more frequently, after a few changes instead of waiting for all changes to the application.

Unsuccessful Log On

Scenario:

When logging on to Oracle Hyperion Planning Web client, this error displays: "Unsuccessful login. Please try again."

Solutions:

Ensure that you are using a valid username and password.

Ensure that Oracle Essbase server and Oracle Hyperion Shared Services are running.

Review the error log. (See *Oracle Enterprise Performance Management System Installation and Configuration Troubleshooting Guide*.)

If the Log On button does not appear at logon, adjust the security settings in your Web browser, and add the Planning server name as a Trusted Site. For example, in Internet Explorer, select **Tools**, then **Internet Options**, then **Security**, then **Trusted Sites**, and then **Sites** and add `http://servername` to the trusted sites zone.

A

Naming Restrictions

Restrictions for Applications and Databases

When naming an application, follow these rules:

- Use no more than eight characters for an application's name.
- Do not use spaces.
- Do not use these special characters:
 - asterisks
 - backslashes
 - brackets
 - colons
 - commas
 - equal signs
 - greater than signs
 - less than signs
 - periods
 - plus signs
 - question marks
 - quotation marks (double and single)
 - semicolons
 - slashes
 - tabs
 - vertical bars
- For application names in relational database environments, do not use extended characters (except for underscores).
- For aggregate storage databases, do not use DEFAULT, LOG, METADATA, or TEMP as application names.

Enter names in the preferred case. Application names are created exactly as entered.

For detailed information on creating Oracle Essbase applications and databases, see the Essbase product documentation.

Restrictions for Dimensions, Members, Aliases, and Forms

When naming dimensions, members, and aliases, follow these rules:

- For non-Unicode mode dimensions, members, or aliases, use no more than 80 bytes. For Unicode-mode dimensions, members, or aliases, use no more than 80 characters.
- Do not use HTML tags in member names, dimension names, aliases, and descriptions.
- Do not use quotation marks, brackets, backslashes, or tabs. Brackets are permitted but not recommended in block storage outlines. They cause errors when converting to aggregate storage outlines.
- To begin dimension or member names, do not use these characters:
 - at signs
 - backslashes
 - brackets
 - commas
 - dashes, hyphens, or minus signs
 - equal signs
 - less than signs
 - parentheses
 - periods
 - plus signs
 - quotation marks
 - underscores
 - vertical bars
- Do not place spaces at the beginning or end of names. Such spaces are ignored.
- Do not use forward slashes in member names.
- For time periods in custom calendars, do not use spaces in prefixes.
- Do not use these words as dimension or member names:
 - Calculation script commands, operators, and keywords.
For a list of commands, see the Oracle Essbase product documentation.
 - Report Writer commands; see the *Oracle Essbase Technical Reference*.
 - Function names and function arguments.
 - Names of other dimensions and members (unless the member is shared), and generation names, level names, and aliases in the database.
 - If Dynamic Time Series is enabled, do not use History, Year, Season, Period, Quarter, Month, Week, or Day.
 - These words:
 - * ALL
 - * AND
 - * ASSIGN
 - * AVERAGE
 - * CALC
 - * CALCMBR

- * COPYFORWARD
- * CROSSDIM
- * CURMBRNAME
- * DIM
- * DIMNAME
- * DIV
- * DYNAMIC
- * EMPTYPARM
- * EQ
- * EQOP
- * EXCEPT
- * EXP
- * EXPERROR
- * FLOAT
- * FUNCTION
- * GE
- * GEN
- * GENRANGE
- * GROUP
- * GT
- * ID
- * IDERROR
- * INTEGER
- * LE
- * LEVELRANGE
- * LOCAL
- * LOOPBLOCK
- * LOOPPARMS
- * LT
- * MBR
- * MBRNAME
- * MBRONLY
- * MINUS
- * MISSING
- * MUL
- * MULOP
- * NE
- * NON

- * NONINPUT
- * NOT
- * OR
- * PAREN
- * PARENPARAM
- * PERCENT
- * PLUS
- * RELOP
- * SET
- * SKIPBOTH
- * SKIPMISSING
- * SKIPNONE
- * SKIPZERO
- * STATUS
- * TO
- * TOLOCALRATE
- * TRAILMISSING
- * TRAILSUM
- * TYPE
- * UMINUS
- * UPPER
- * VARORXMBR
- * XMBRONLY
- * \$\$UNIVERSE\$\$
- * #MISSING
- * #MI

Dimension and Member Names in Calculation Scripts, Report Scripts, Formulas, Filters, and Substitution Variables

In substitution variable values, calculation scripts, report scripts, filter definitions, partition definitions, or formulas, you must enclose member names in brackets ([]) when used within MDX statements and in quotation marks (" ") for block storage databases, in these situations:

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

Character	Description	Character	Description
&	ampersand	>	greater than sign
*	asterisk	<	less than sign

Character	Description	Character	Description
@	at sign	()	parentheses
\	backslash	%	percent sign
{ }	braces	.	period
:	colon	+	plus sign
,	comma	;	semicolon
-	dash, hyphen, or minus	/	slash
!	exclamation point	~	tilde
=	equal sign		

In calculation scripts and formulas, enclose member names that are also Oracle Essbase keywords in quotation marks (") for block storage databases, and in brackets ([]) for aggregate storage databases including these member names:

BEGIN DOUBLE ELSE END FUNCTION GLOBAL IF MACRO
MEMBER RANGE RETURN STRING THEN

Enclose in quotation marks names, in calculation scripts, report scripts, or formulas, that contain or start with spaces or these characters: + - * / () : , @ ;) { } [] <

Also enclose in quotation marks names that start with numbers. For example:

```
@SUM(ac1, "12345");
```

Restrictions for User Names

User names can have up to 80 characters.

B

Form Formula Functions

Using Formulas and Formula Functions


After you create a formula row or column, define the formula using the **Segment Properties** pane. Formulas include grid references, mathematical operators, and, optionally, mathematical functions. For a complete list, see [Formula Functions](#).

 **Note:**

Oracle Hyperion Planning requires level 0 members that are Dynamic Calc to have a member formula. For Dynamic Calc members that do not have a formula, Planning inserts a semicolon (;) when refreshed; the semicolon is visible in the Formula field in Oracle Essbase.


Creating Formulas

To create formulas:

1. Select the row or column for which to associate the formula.
2. If it does not automatically expand, select **Segment Properties**.
3. In Formula, enter the name for the formula and then click .
4. In the **Formula** box that is displayed, select the operation or function that the formula will perform, such as COUNT(), MAX(), and IfThen(). See [Formula Functions](#).
5. Click **Validate** to ensure that the formula does not contain any errors.

Editing Formulas

To edit formulas:

1. Open a form with a formula row or column.
2. In **Layout**, select a numbered formula row or column.
3. Use **Segment Properties** to modify the formula properties.
4. Click  to change the operation or function performed.
5. Save the form.

Deleting Formulas

To delete a formula:

1. Select the formula row or column.
2. Click the **Delete** button.
3. To verify formula deletion:
 - a. Click the checkmark in the formula bar.
 - b. Click another cell within the grid to reset the formula bar.
 - c. Click the cell from which you deleted the formula to verify deletion.

Formula Functions

This section defines the mathematical functions available for creating formulas for form formula rows and columns. To insert formula rows and columns in forms, see [Adding Formula Rows and Columns](#).

The syntax for mathematical functions:

```
FunctionName (arguments)
```

Table B-1 Mathematical Function Syntax

Variable	Description
<i>FunctionName</i>	The name of a mathematical function
<i>arguments</i>	A numeric value, a row, column, or cell reference, or an embedded function

Table B-2 Mathematical Functions

Function	Description
Abs	Returns the absolute value of numeric values or references
Average	Returns the average of a group of numeric values or references
AverageA	Returns the average of a group of numeric values or references. The calculation includes #MISSING cells only for rows or columns that are not suppressed.
Count	Returns the number of values in a group of numeric values or references
CountA	Returns the number of values in a group of numeric values or references. The calculation includes #MISSING cells only for rows or columns that are not suppressed.
Difference	Returns the absolute value of a numeric value or reference subtracted from another numeric value or reference
Eval	Evaluates an expression. Eval is useful for embedding expressions as function arguments

Table B-2 (Cont.) Mathematical Functions

Function	Description
IfThen, If	Returns one value if a condition equals true, and another value if a specified condition equals false
Max	Returns the maximum value from a group of numeric values or references
Min	Returns the minimum value from a group of numeric values or references
Mod	Returns the remainder, modulus, from a division formula
PercentOfTotal	Returns the result of a numeric value or reference divided by another numeric value or reference, multiplied by 100
Pi	Returns the number 3.14159265358979, to 15 digits
Product	Multiplies all numbers or references and returns the product
Random	Returns a random number between 0.0 and 1.0
Rank	Returns the highest or lowest value of a specified column or row
Round	Rounds a number up or down by specified digits
Sqrt	Returns the square root of a numeric value, row, column, or cell
Sum	Returns the sum of a group of numeric values or references
Truncate / Trunc	Removes the specified number of digits from numeric values
Variance / Var	Evaluates the difference between the specified values based on the account type for the current account
VariancePercent / VarPer	Evaluates the percentage difference between the specified values based on account type for the current account

Arguments

Mathematical functions accept numeric values, row, column, or cell references, or embedded functions as arguments. There are four argument types:

- Numeric
- Property
- Row, column, or cell reference
- Embedded functions

Numeric Arguments

The syntax for a numeric argument is

```
(numeral1, numeral2, ... numeraln)
```

where numerals 1 through *n* are any numbers including decimals and negative values. For example, the expression `Average(10,20,30)` returns the value 20.

Row, Column, or Cell Reference Arguments

The row, column, or cell argument identifies a row, column, or cell in a grid. The syntax:


```
FunctionName (GridName.GridElement[segment (range)].Property)
```

Table B-3 Argument Components

Argument	Description
<i>GridName</i>	The form name. For example: Difference (<code>grid1.row[5]</code> , <code>grid1.row[6]</code>) returns the difference of two rows on form <code>grid1</code> . Optional. If <i>GridName</i> is not specified, the default is the name of the current form.
<i>GridElement</i>	One of the following keywords: <code>row</code> , <code>col</code> , <code>column</code> , or <code>cell</code> . For example, <code>Max(row[1], row[2], row[3])</code> returns the maximum value of three rows. <i>GridElement</i> is optional. However, a cell reference requires row and column segment identifiers. For example, <code>cell[2, A]</code> and <code>[2, A]</code> both refer to the cell that is the intersection between row 2 and column A. The keyword <code>cell</code> is optional. Cell references can use the <code>[row, col]</code> syntax or <code>[col, row]</code> syntax. Optional. If <i>GridElement</i> is not specified, letters represent columns and numbers represent rows; for example: <code>Max ([1, A], [2, A], [3, A])</code> refers to rows 1, 2 and 3 of column A.
<i>segment</i>	A row, column, or cell reference number. For an expanded row or column, you must specify the segment. For example, <code>row[2]</code> addresses row segment 2. Segments are enclosed in square brackets <code>[]</code> . Required.

Table B-3 (Cont.) Argument Components

Argument	Description
<i>range</i>	<p>The rows, columns, or cell that are expanded from the specified segment. If <i>range</i> is specified, the system calculates the formula using only the specified range. For example, <code>row[2(3:5)]</code> uses only the third through fifth rows of expanded segment 2. Optional. When <i>range</i> is not provided, all expanded cells are used.</p>
<i>property</i>	<p>One of these keywords: <code>average</code>, <code>averageA</code>, <code>count</code>, <code>countA</code>, <code>max</code>, <code>min</code>, <code>product</code>, or <code>sum</code>. The property specifies how to aggregate the specified expanded rows, columns, or cells.</p> <p>Oracle recommends that <i>property</i> is <i>not</i> specified when a reference is an argument. By not specifying the property, the function calculates the reference in the most appropriate way. For example, the following expression returns the average of the cells within rows 1 and 2:</p> <pre>Average(row[1], row[2])</pre> <p>In contrast, the following example first calculates the average of <code>row[1]</code>, then the average of <code>row[2]</code>, adds these two results, then divides by 2:</p> <pre>Average(row[1].average, row[2].average)</pre> <p>The default property for a row, column, or cell reference is <code>sum</code>. For example, <code>row[2]</code> is equivalent to <code>Sum(row[2])</code>.</p>

 **Note:**
If a segment expands to only one row or column, do not use the range argument.

Because *segment* is the only required part of a reference, the following references are the same:

```
Grid1.row[1].sum
```

```
[1]
```

`AverageA` and `CountA` include `#MISSING` cells in the calculation. For example, if row 1 is a segment row that expands to `Qtr1 = 100`, `Qtr2 = 200`, `Qtr3 = #MISSING`, and `Qtr4 = 400`, the following function returns the value four (4):

```
row[1].CountA
```

All other functions exclude #MISSING data cells. For example, the previous example of row 1 that expands to Qtr1 = 100, Qtr2 = 200, Qtr3 = #MISSING, and Qtr4 = 400, returns three in this example:

```
row[1].Count
```

Property Arguments

Property arguments consolidate expanded references to a single value that is then used in the calculation. Use property arguments to perform calculations on an aggregate row, column, or cell. The two types of property arguments are:

- Aggregate Property Argument (see [Aggregate Property Argument](#))
- Reference Property Argument (see [Reference Property Argument](#))

Aggregate Property Argument

An aggregate row, column, or cell contains multiple rows, columns, or cells, respectively.

The aggregate property argument is the last argument in the following mathematical function syntax:

```
FunctionName(GridName.Gridelement[segment(range)].property)
```

You apply the following aggregate properties to a row, column, or cell reference.

Table B-4 Aggregate Properties

Properties	Description
Average	Returns the average of a row, column, or cell. The calculation excludes #MISSING and #ERROR values.
AverageA	Returns the average of a row, column, or cell. The calculation includes #MISSING and #ERROR values.
Count	Returns the number of values in a row, column, or cell. The calculation excludes #MISSING and #ERROR values.
CountA	Returns the number of values in a row, column, or cell. The calculation treats #MISSING and #ERROR values as zero (0).
Max	Returns the maximum value of a row, column, or cell
Min	Returns the minimum value of a row, column, or cell
Product	Returns the product of rows or columns
Sum	Returns the sum of a row, column, or cell

When used as a mathematical function argument, the default for property is the same as the function. In the following example the default property is Average:

```
Average(row[2])
```

When not used as a mathematical function argument, the default for property is sum. In the following example the default property is the sum of an aggregate row:

```
row[2]
```

Reference Property Argument

A reference property argument specifies how to treat formula reference results and is used in conjunction with the other properties.

There is one reference property argument: `IfNonNumber/IFFN`.

`IfNonNumber` specifies a replacement of `#MISSING` and `#ERROR` values with a specific numeric value. The syntax:

```
AXIS[segment(range)].IfNonNumber(arg).AggregateProperty
```

Argument	Description
AXIS	One of these keywords: row, column, or cell Optional
Segment(range)	Indicates any valid axis reference, such as a row number, column letter
IfNonNumber	Indicates how to treat missing or error data within the Axis Ref
(arg)	Indicates what number to use if missing or error data is encountered within the AxisRef
AggregateProperty	The aggregate function is used for aggregate segments Optional

For example:

If `cell[1,A] = 3` and

`cell[1,B] = #MISSING`,

The expression:

```
cell[1,A] / cell[1,B]
```

returns `#ERROR`.

The expression:

```
cell[1,A] / cell[1,B].ifNonnumber(1)
```

replaces `cell[1,B]` with 1 and returns a 3.

**Note:**

If you use suppression for #MISSING in a grid, and the grid contains a formula row or column that uses the IfNonNumber property, #MISSING remains suppressed.

Embedded Functions as Arguments

You can embed functions as arguments within a function.

Example:

In this example, the function Average is embedded in the function Sum:

```
sum(row[3:5], avg(row[4:6], 40, 50), row[7; 9], 70, 80)
```

- Row segments 3, 4 and 5
- The average of row segments 4, 5 and 6, with the numbers 40 and 50
- Row segments 7 and 9
- The numbers 70 and 80

Abs

Abs is a mathematical function that returns the absolute value of a numeric value, row, column, or cell. The absolute value of a number is that number without regard to sign. A negative number becomes positive, while a positive number does not change. The function syntax:

```
Abs (argument)
```

where *argument* is one of the following:

Argument	Description
numeric	A numeric value. For example, Abs (- 20) returns the value 20. Numeric values can include decimals and negative values.
row, column, or cell reference	A pointer to a row, column, or cell within a grid. References can be specified in several ways. The reference syntax: <code>GridName.GridElement[segment (range)].Property.</code>
function	An embedded function

Examples:

The following expression includes a numeric argument and returns the value 30:

```
Abs ( -30 )
```

The following example returns the absolute value of the value in row 1:

```
Abs(row[1])
```

The following example calculates the absolute value of the sum of column E:

```
Abs(column[E].sum)
```

The following example points to expanded rows 1 through 3 within design segment 3 of the form Grid1:

```
Abs(Grid1.row[3(1:3)])
```

Average

Average is a mathematical function that returns the average of a group of numeric values, rows, columns, or cells. Average excludes #MISSING and #ERROR cells when obtaining the average.



Note:

The calculation does not include missing values regardless of whether they are suppressed or not.

The function syntax:

```
Average(arguments) or Avg(arguments)
```

where *arguments* is one or more of the following:

Argument	Description
numeric	A numeric value. For example, Average (10, 20, 30) returns the value 20. Numeric values can include decimals and negative values.
row, column, or cell reference	A pointer to a row, column, or cell within a grid. References can be specified in several ways. The reference syntax: GridName.GridElement[segment(range)].Property For example Avg(Grid1.row[4(3:5)]) returns the average of form grid1, row 4, range 3 through 5.
function	An embedded function

Examples:

The following expression returns the value 20:

```
Avg(10, 30, 20)
```

The following example returns the average of all numbers that are part of three aggregate rows:

```
Average( row[1], row[6], row[8] )
```

The following example calculates the average of three aggregate columns; E, G, and I. The calculation produces three numbers and calculates the average of the three numbers:

```
Avg(column[E].avg, column[G].avg, column[I].avg)
```

The following example calculates the average of aggregate row 3 and divides the average by 100:

```
Avg(row[3])/100
```

AverageA

AverageA is a mathematical function that returns the average of a group of numeric values, rows, columns, or cells. AverageA includes #MISSING and #ERROR cells, which are treated as zero values when obtaining the average.



Note:

#MISSING and #ERROR are included only for rows or columns that are not suppressed.

The function syntax:

```
AverageA(arguments) or AvgA(arguments)
```

where *arguments* is one or more of these arguments:

Argument	Description
numeric	A numeric value. For example, AverageA (10, 20, 30) returns the value 20. Numeric values can include decimals and negative values.
row, column, or cell reference	A pointer to a row, column, or cell. References can be specified in several ways. The reference syntax: GridName.GridElement[segment(range)]. Property. For example, AvgA(Grid1.row[4(3:5)]) returns the average of form grid1, row segment 4, range 3 through 5.
function	An embedded function

Example

In the following example, a grid has 4 rows with the values 10, 20, 30, and, #ERROR. The following formula in the fifth row returns the value 15:

```
AverageA([1:4])
```

Count

Count is a mathematical function that returns the number of values in a group of numeric values, rows, columns, or cells. Count excludes #MISSING and #ERROR when obtaining the count. The function syntax:

```
Count (arguments)
```

where *arguments* is one or more of these arguments:

Argument	Description
numeric	A numeric value. For example, Count (10, 20, 30) returns the value 3. Numeric values can include decimals and negative values.
row, column, or cell reference	A pointer to a row, column, or cell within a grid. References can be specified in several ways. The reference syntax: GridName.GridElement[segment(range)].Property
function	An embedded function

Examples:

The following example returns the count of three rows, 1, 6, and 8:

```
Count(row[1], row[6], row[8])
```

The following example returns the count of 3 columns:

```
Count(column[E], column[G], column[I] )
```

The following example calculates the count of the cell located at row 4, column D:

```
Count(cell[D,4])
```

The following example calculates the count of aggregate row 3 in grid 5:

```
Count(grid1.row[3])
```

CountA

CountA is a mathematical function that returns the number of values in a group of numeric values, rows, columns, or cells. CountA includes #MISSING and #ERROR cells when obtaining the count only for rows or columns that are not suppressed. The function syntax:

CountA(arguments)

where *arguments* is one or more of these arguments:

Argument	Description
numeric	A numeric value. For example, CountA(10,20,30,50) returns the value 4. Numeric values can include decimals and negative values.
row, column, or cell reference	A pointer to a row, column, or cell within a grid. References can be specified in several ways. The reference syntax: CountA(GridName.GridElement[segment(range)].property)
function	An embedded function

Example

In the following example, if a grid has 4 rows with the values 10, 20, 30, and, #ERROR. The following formula in the fifth row returns the count of four rows:

```
CountA([1:4])
```

The following example returns the count of four rows:

```
CountA(row[1], row[6], row[8] row[where data yields #ERROR])
```

Difference

Difference is a mathematical function that returns the absolute value of the difference of a numeric value, row, or column subtracted from another numeric value, row, or column. This is also known as the variance. The function syntax:

Difference(arg1, arg2)

where *arg2* is subtracted from *arg1* and are one or more of the following arguments:

Argument	Description
numeric	A numeric value. For example, Difference(3, 5) returns the absolute value 2. Numeric values can include decimals and negative values.

Argument	Description
row, column, or reference	A pointer to a row, column, or cell within a grid. References can be specified in several ways. The reference syntax: <code>GridName.GridElement[segment (range)].Property</code> The following example returns the difference of two rows in form grid1: <code>Difference(grid1.row[1], grid1.row[6])</code>
function	An embedded function

 **Note:**

The Difference function returns the absolute value of arg2 subtracted from arg1, whereas the minus sign in subtraction negates a number.

Examples:

The following example returns the absolute value of 8:

```
Difference(3, -5)
```

The following example calculates the difference of two aggregate columns:

```
Difference( column[E], column[G] )
```

 **Note:**

You can type the text label "Difference" or "Variance."

Eval

Eval is a mathematical function that evaluates an expression. You use Eval as an embedded function argument to consolidate multiple expressions into one expression. The function syntax:

Eval(expression)

where *expression* is one or more of these arguments:

Argument	Description
numeric	A numeric value. Numeric values can include decimals and negative values.

Argument	Description
row, column, or reference	A pointer to a row, column, or cell within a grid. References can be specified in several ways. The reference syntax: <code>GridName.GridElement[segment(range)].Property</code>
function	An embedded function
operators	Use any of the supported arithmetic operators (+, -, *, /, ^, %, and so on).

Example

The following example divides row 1 by row 2 and then rounds the data to four places:

```
Round(Eval([1]/[2]),4)
```

IfThen, If

IfThen is a conditional function that returns a value when the condition equals True, and another value when the condition equals False.

The function syntax:

```
IfThen(Condition, TrueParameter, FalseParameter)
```

- *Condition* is a logical expression that evaluates to true or false. Full conditional logic can be used as well as complex Boolean operators (And, Not, and Or). A *condition* can also test for #MISSING and #ERROR values. See the following table for a list of valid conditional operators.
- *TrueParameter* and *FalseParameter* are any valid expression that are evaluated based on the outcome of the condition.

The following table describes the conditional operators that are fully supported. Alternate syntax is listed wherever it is supported by a conditional operator.

Table B-5 Conditional Operators

Conditional Operator	Syntax	Logic
Equal To	<i>expression = expression</i>	Tests whether the left expression is equal to the right expression.

 **Note:**

The routine that evaluates the condition does not consider any rounding. If rounding is required, use the Round function.

Example:

1
=

4
R
e
t
u
r
n
s
a
s
e

Table B-5 (Cont.) Conditional Operators


Conditional Operator	Syntax	Logic
Greater Than	<i>expression</i> > <i>expression</i>	Tests whether the left expression is greater than the right expression. Example:

1
>
4

R
e
t
u
r
n
s
a
s
e

Table B-5 (Cont.) Conditional Operators

Conditional Operator	Syntax	Logic
Greater Than or Equal To	<i>expression</i> >= <i>expression</i>	Tests whether the left expression is greater than or equal to the right expression.

 **Note:**

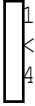
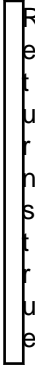
The correct syntax is ">=". The syntax "=>" is not supported.

Example:

1
>
=

R
E
C
E
I
V
E
S
E
S
E

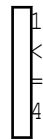
Table B-5 (Cont.) Conditional Operators

Conditional Operator	Syntax	Logic
Less Than	<i>expression < expression</i>	Tests whether the left expression is less than the right expression. Example:  
Less Than or Equal To	<i>expression <= expression</i>	Tests whether the left expression is less than or equal to the right expression.

 **Note:**

The correct syntax is "<=". The syntax "<=" is not supported.

Example:



Returns true

Table B-5 (Cont.) Conditional Operators

Conditional Operator	Syntax	Logic
Not Equal To	$expression \lt \gt expression$ $expression \neq expression$	Tests whether the left expression is not equal to the right expression.



Note:

The routine that evaluates the condition does not consider any rounding. If rounding is required, use the Round function.

Example:

1
<<
>>
14

R
e
c
u
r
s
e
c
u
r
e

1
1
14

R
e
c
u
r
e

Table B-5 (Cont.) Conditional Operators

Conditional Operator	Syntax	Logic
		true

Table B-5 (Cont.) Conditional Operators

Conditional Operator	Syntax	Logic
IsMissing	IsMissing (<i>reference</i>) IsMiss (<i>reference</i>)	Tests whether the reference contains a #MISSING result.

 **Note:**

If the reference is an expanded row or column, then all resulting cells must be #MISSING in order for the condition to be true.

Example:

IsMissing (

Reference

Table B-5 (Cont.) Conditional Operators

Conditional Operator	Syntax	Logic
		b w 1 h a s a # M S S N G v a u e

Table B-5 (Cont.) Conditional Operators

Conditional Operator	Syntax	Logic
IsError	IsError (<i>reference</i>) IsErr (<i>reference</i>)	Tests whether the reference contains an #ERROR result.

 **Note:**

If the reference is an expanded row or column, all resulting cells must be #ERROR in order for the condition to be true. Only formula rows and columns can result in #ERROR.

Example:

ISERROR (COLUMN (1:2))

TRUE

Table B-5 (Cont.) Conditional Operators

Conditional Operator	Syntax	Logic

e
f
r
o
w
2
n
a
s
a

E
R
R
O
R
v
a
r
i
a
n
t
e

Table B-5 (Cont.) Conditional Operators

Conditional Operator	Syntax	Logic
IsNonNumeric	IsNN (<i>reference</i>) IsNonNumerid (<i>reference</i>) IfNN (<i>reference</i>) IfNonNumber (<i>reference</i>)	Tests whether the reference contains a #MISSING or #ERROR results.

 **Note:**

If the reference is an expanded row or column, all resulting cells must be #MISSING and/or #ERROR in order for the condition to be true.

Example:

ISNN(
[3])

Returns true if row 3

Table B-5 (Cont.) Conditional Operators

Conditional Operator	Syntax	Logic
Parenthesis	<i>(condition)</i>	<p>Used to group a condition. Used mostly for visual clarity.</p> <p>Example:</p>

a
s
a

M
S
S
Z
G
o
t

E
R
R
O
R
v
a
r
i
a
n
t

(
t
v
t
)

R
e
t
u
r
n
s
e
r
v
e
r
e
r
s
e

Table B-6 Conditional Operators

Complex Conditions	Syntax	Logic
And	<p>(<i>condition</i> AND <i>condition</i>) (<i>condition</i> & <i>condition</i>)</p>	<p>Complex condition used to compare two conditions. Returns true if all conditions result in true. Example:</p>

(1
>
4
A
N
O
5
>
S
)

R
e
s
u
l
t
s
e
s
s
e
s

Table B-6 (Cont.) Conditional Operators

Complex Conditions	Syntax	Logic
Not	NOT (<i>condition</i>) ! (<i>condition</i>)	Used to negate the result by reversing the result of the condition. Example:

NOT

! (condition)

Table B-6 (Cont.) Conditional Operators

Complex Conditions	Syntax	Logic
Or	<p><i>(condition</i> OR <i>condition)</i></p> <p><i>(condition</i> <i>condition)</i></p>	<p>Complex condition used to compare two conditions. Returns true if any of the conditions result in true.</p> <p>Example:</p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin-left: auto; margin-right: auto;"> <p>(1 4 O R 5 2) R E T U R N S T R U E</p> </div>

Notes on Conditions

- *Expression*—Any valid formula expression. The expression can be any combination of a constant (integer or real number), a reference, or another function.
- *Reference*—Any valid reference; thus the IFNN reference property can be used as part of the reference.
- *Condition*—Any valid condition applied to the complex conditions And, Not, and Or. Those operators can have embedded conditions.



Note:

And, Not, and Or operators require surrounding parentheses.

- When any *expression* within the condition returns an #ERROR or #MISSING value, the If function returns #MISSING or #ERROR. This does not apply when you use the IsMissing, IsError, or IsNonNumeric conditions.

Complex Conditions

Complex conditions And, Or, and Not are fully supported. However, they must be surrounded by parentheses.

Valid example:

```
If ( ([A] > [B] and [A] > 1000), [A], [B])
```

Invalid example:

```
If ( [A] > [B] and [A] > 1000, [A], [B])
```

Max

Max function is a mathematical function that returns the maximum value in a group of numeric values, rows, columns, or cells. The function syntax:

```
Max (arguments)
```

where *arguments* is one or more of these arguments:

Argument	Description
numeric	A numeric value. For example, Max (10, 20, 30) returns the value 30. Numeric values can include decimals and negative values.
row, column, or cell reference	A pointer to a row, column, or cell within a grid. References can be specified in several ways. The reference syntax: GridName.GridElement[segment(range)].Property.
function	An embedded function

Examples:

The following example returns the maximum value in rows 1, 6, and 8:

```
Max(row[1], row[6], row[8])
```

The following example calculates the maximum of the sums of aggregate rows:

```
Max(row[1].sum, row[2].sum, row[3].sum)
```

Min

Min is a mathematical function that returns the minimum value in a group of numeric values, rows, columns, or cells. The function syntax:

Min (*arguments*)

where *arguments* is one or more of these arguments:

Argument	Description
numeric	A numeric value. For example, Min (10, 20, 30) returns the value 10. Numeric values can include decimals and negative values.
row, column, or cell reference	A pointer to a row, column, or cell within a grid. References can be specified in several ways. The reference syntax: GridName.GridElement[segment(range)].Property
function	An embedded function

Examples:

The following example returns the minimum value in rows 1, 6, and 8:

```
Min (row[1], row[6], row[8])
```

The following example calculates the minimum of the sums of aggregate rows:

```
Min(row[1].sum, row[2].sum, row[3].sum)
```

Mod

Mod is a mathematical function that returns the remainder, or modulus, from a division. The function syntax:

Mod (*arg1*, *arg2*)

where *arg2* is the divisor and *arg1* and *arg2* are one of these arguments:

Argument	Description
numeric	A numeric value. For example, Mod (6, 4) returns the value 2. Numeric values can include decimals and negative values.
row, column, or cell reference	A pointer to a row, column, or cell within a grid. References can be specified in several ways. The reference syntax: GridName.GridElement[segment(range)].Property
function	An embedded function

Example:

The following example divides 10 by 5 and returns the remainder of 0:

```
Mod (10,5) = 0
```

PercentOfTotal

PercentOfTotal is a mathematical function that returns the result of a numeric value, row, column, or cell divided by another numeric value, row, column, or cell which is multiplied by 100. The function syntax:

```
PercentOfTotal (arg1, arg2)
```

- where *arg1* is a component of the running total (*arg2*). Normally, this is a row or column reference.
- where *arg2* is the running total relative to *arg1*. Normally this is a cell reference containing the grand total.
- *arg1* is divided by *arg2*, with the result multiplied by 100. *Arg1* and *arg2* are one or more of these arguments:

Argument	Description
numeric	A numeric value. For example, PercentofTotal (100, 20) returns the value 500. Numeric values can include decimals and negative values.
row, column, or cell reference	A pointer to a row, column, or cell within a grid. References can be specified in several ways. The reference syntax: GridName.GridElement[segment (range)].Property
function	An embedded function

 **Note:**

This function requires two arguments.

Examples:

The following example returns the value of 5 percent.

```
PercentofTotal (20, 400)
```

The following example divides the value of each cell in column A by the Total Mkt value in cell A5, multiplies the result by 100, and displays the resulting PercentOfTotal in column B. The formula:

```
PercentOfTotal ([A], [A,5]),
```

Using the above example, the following table shows the PercentOfTotal results in column B:

		A	B
1		Sales	% Total
2	Mkt1	60	20%
3	Mkt2	120	40%
4	Mkt3	120	40%
5	Total Mkt	300	100%



Tip:

You enter the formula by clicking on the header for Column B and using the formula bar.

Pi

Pi is a mathematical function that returns the number 3.14159265358979, the mathematical constant, accurate to 15 digits. Pi is the ratio of the circumference of a circle to its diameter. The function syntax:

PI ()

Example:

The following example returns row 3 divided by the product of Pi and 2:

```
row[3] / (PI() * 2)
```

Product

Product is a mathematical function that multiplies all numbers or references and returns the product. The function syntax:

Product(arguments)

where *arguments* is one or more of these arguments:

Argument	Description
numeric	A numeric value. For example, Product(2, 20) returns the value 40. Numeric values can include decimals and negative values.
row, column, or cell reference	A pointer to a row, column, or cell within a grid. References can be specified in several ways. The reference syntax: GridName.GridElement[segment(range)].property
function	An embedded function

Example:

The following example returns 40:

```
Product(2,20)
```

Random

Random is a mathematical function that returns a random number between 0.0 and 1.0. The function syntax:

```
Random()
```

Example:

The following example returns a random number between 0.0 and 1.0 and multiplies it by 1000:

```
Random() * 1000
```

Rank


Rank is a financial function that provides a rank value for a value in a specified range. The Rank function is processed by Oracle Hyperion Financial Reporting and does not depend on the database connection. The function syntax:

```
Rank([Reference], Order)
```

```
Rank([Reference], Order, Unique)
```

Argument	Description
<i>Reference</i>	<p>The range of cells, rows, or columns to rank, with letters identifying columns and numbers identifying rows. For example, specify [A,1:5] to rank the values for rows 1 through 5 in column A.</p> <p>You can use the .ifNN property with a range of cells to assign numbers to any cells with nonnumeric values so that those cells can be ranked. For example, you can use .ifNN(-1) to assign the value -1 to any cell with a missing value.</p>

Argument	Description
<i>Order</i>	<p>Indicates the order by which the values are ranked. The lowest value ranked in ascending order receives a rank result of 1. The largest value ranked in descending order receives a rank result of 1. The order can be indicated by any of the following keywords or values:</p> <ul style="list-style-type: none"> • Ascending • Descending • Asc • Des • Desc • 1 (the number 1 is the same as "ascending") • 0 (zero is the same as "descending") <p>The keywords are not case-sensitive.</p>
<i>Unique (optional)</i>	<p>Optional. A Boolean keyword indicating how to treat equal values in the Reference parameter where:</p> <ul style="list-style-type: none"> • false (or omitted) — equal values receive the same ranking ; ranked results may be duplicated • true — equal values receive a unique ranking; there are no duplicate rankings. Values in the Reference parameter are ranked on a first come, first ranked basis. For example, if values in rows 2 and 5 are equal, the value in row 2 is ranked before that of row 5.

 **Note:**
Do not enclose the number or keyword indicating order in quotation marks.

Examples

This formula in column B ranks the values in rows 1 through 5 in column A in descending order:

```
Rank([A,1:5], descending)
```

The result might be as follows:

Product	East	Rank
Cola	16	2
Fruit Drinks	23	1
Beer	16	2
Diet	missing	missing

Product	East	Rank
Root Beer	0	4

When two values are equal, they receive the same rank value. In the example above, Cola and Beer have the same value and therefore the same rank.

This formula in column B assigns the value of -1 to any nonnumeric value so it can be ranked:

```
Rank([A,1:5].ifNN(-1), descending)
```

In the following result, the missing value now has a rank of 5:

Product	East	Rank
Cola	16	2
Fruit Drinks	23	1
Beer	16	2
Diet	missing	5
Root Beer	0	4

Example:

The following example builds on the previous example explaining how the unique parameter affects the results:

This formula in column B assigns the value of -1 to any nonnumeric value so it can be ranked, and also indicates that each ranking should be unique:

```
Rank([A,1:5].ifNN(-1), descending, true)
```

In the following result, the missing value now has a rank of 5, and Beer has a value of 3 (even though it has the same data value as Cola):

Product	East	Rank
Cola	16	2
Fruit Drinks	23	1
Beer	16	3
Diet	missing	5
Root Beer	0	4

Round

Round is a mathematical function that rounds a number up or down by the specified digits. The function syntax:

Round (*arg1*, *integer*)

where *arg1* is one or more of these arguments:

Argument	Description
<i>numeric</i>	A numeric value. For example, Round(81.3987,3) returns the value 81.399. Numeric values can include decimals and negative values.
row, column, or cell reference	A pointer to a row, column, or cell within a grid. References can be specified in several ways. The reference syntax: GridName.GridElement[segment(range)].property
<i>function</i>	An embedded function

Integer specifies the number of digits to which you want to round the number:

- If *integer* is greater than zero, the number rounds to the specified number of decimal places.
- If *integer* is zero, the number rounds to the nearest integer.
- If *integer* is less than zero, the number is rounded to the left of the decimal point.

Examples:

The following example rounds to three decimals:

Round(3594.5567, 3) = 3594.557

The following example rounds to the nearest integer:

Round(3594.5567, 0) = 3595

The following example rounds to the thousands. This is also known as scaling:

Round(3594.5567, -3) = 4000

Sqrt

Sqrt is a mathematical function that returns the square root of a numeric value, row, column, or cell. The syntax for the Sqrt function:

Sqrt (*argument*)

where *argument* is one of these arguments:

Argument	Description
<i>numeric</i>	A numeric value. For example, Sqrt(100) returns the value 10. Numeric values can include decimals and negative values.
row, column, or cell reference	A pointer to a row, column, or cell within a grid. References can be specified in several ways. The reference syntax: GridName.GridElement[segment(range)].property

Argument	Description
<i>function</i>	An embedded function

Example

The following example returns the value of 4:

```
Sqrt(16)
```

Sum

Sum is a mathematical function that returns the summation of a group of numeric values, rows, columns, or cells.

The syntax for the Sum function:

Sum (arguments)

where *arguments* is one or more of these arguments:

Argument	Description
<i>numeric</i>	A numeric value. For example, Sum(10, 20, 30) returns the value 60. Numeric values can include decimals and negative values.
row, column, or cell reference	A pointer to a row, column, or cell within a grid. References can be specified in several ways. The reference syntax: GridName.GridElement[segment(range)].property
<i>function</i>	An embedded function

Examples:

- The following example returns the value 30:

```
sum(10,20)
```

- The following example returns the sum of three rows:

```
sum(row[1],row[6],row[8])
```

- The following example calculates the sum of three aggregate columns:

```
sum(column[E], column[G], column[I])
```

When a formula row includes IDESC, the sum includes all of the parents and their descendants.

For example, a form is created with the following rows (and each member has children):

```
IDESC("Mem1"), IDESC("Mem2"), IDESC("Mem3"), IDESC("Mem4")
```

If a formula row is added with the following formula:

```
SUM (ROW[1] ,ROW[2] ,ROW[3] ,ROW[4] )
```

When the form is opened for data entry, the formula row will return the sum of all the parents and their children.

Truncate/Trunc

Truncate is a mathematical function that removes the specified number of digits from numeric values.

Syntax:

```
Trunc (arg1, integer)
```

- where *arg1* is one of these arguments:

Argument	Description
<i>numeric</i>	A numeric value; for example, 234.567.
row, column, or cell reference	A pointer to a row, column, or cell within a grid. References can be specified in several ways. The reference syntax: GridName.GridElement[segment(range)].property
<i>function</i>	An embedded function

- where *integer* specifies the number of digits you want to remove:
 - A positive *integer* determines the number of significant digits that remain to the right of the decimal point.
 - A zero (0) *integer* returns the integer located to the left of the decimal point.
 - A negative *integer* indicates the number of significant digits that are truncated to the left of the decimal point.

Examples:

The following statement uses a positive integer of 2. The first two digits to the right of the decimal point remain, and the following digit is removed:

```
Trunc(234.567, 2) = 234.56
```

The following statement uses a zero (0) integer. All digits to the right of the decimal point are removed:

```
Trunc(234.567, 0) = 234
```

The following statement uses a negative integer of -2. All digits to the right of the decimal point are removed and the last two digits of the integer are truncated.

```
Trunc(234.567, -2) = 200
```



Note:

Formatting previously applied to a cell, column, or row is maintained when you use the Trunc function. The following example shows the results of a Trunc function where the cell value was previously formatted to display three decimal places:
`Trunc(234.567, 0) = 234.000`

Variance/Var

Variance is a financial function that evaluates the difference between the specified values based on account type for the current account. For example, for Expense or Liability accounts, a positive result represents a decrease, so the result appears as a negative number. You can use this function with these UDA account types: Asset, Liability, Equity, Revenue, and Expense.

Syntax:

`Var (reference1, reference2)`

where *reference1* and *reference2* are references to a row, column, or cell that correspond to members of the same Account dimension whose variance results are to be calculated.

Expected Results

The following table describes the expected results when using the Variance function with Accounts.

Table B-7 Expected Results when Using the Variance Function

Column A	Column B	Var ([A] , [B])=0	Var ([A] , [B])>0	Var ([A] , [B])<0
Asset	Asset	0	Returns a positive value	Returns a negative value
Liability	Liability	0	Returns a positive value	Returns a negative value
Equity	Equity	0	Returns a positive value	Returns a negative value
Revenue	Revenue	0	Returns a positive value	Returns a negative value
Expense	Expense	0	Returns a negative value	Returns a positive value

Variance Behavior

- The Variance function expects comparison of the same account type. When you compare two different account types, like Sales & Expense, the Variance function performs the straight math without applying the logic of the account type. For example:

Sales	Expense	Result
-400	100	-500

- When the Variance function is applied to a dimension that is not tagged as an Accounts type, an #ERROR results at runtime.
- #MISSING is treated as zero (0), unless specified differently using the ifnonnumber property.

Examples

The Variance function accepts cell, column, or row references only.

Syntax	Example
Sample syntax referencing a column:	Var ([A], [B])
Sample syntax referencing a row:	Var ([3], [4])
Sample syntax referencing a cell:	Var (Cell [3,A], [3,B])

In this example, the variance between column A (**Actual**) and column B (**Budget**) is calculated as:

Var ([A], [B])

This example produces the following report:

	Year	Product	Market
	Actual	Budget	Variance
	=====	=====	=====
Sales (Income)	400,855	373,080	27,775
COGS (Expense)	179,336	158,940	-20,396

VariancePercent/VarPer

VariancePercent is a financial function that evaluates the difference, in percent, between the specified values based on account type for the current account. For example, for an Income, Flow, Asset, or Balance account, a positive result represents an increase, so the result appears as a positive number. For Expense or Liability accounts, a positive result represents a decrease, so the result appears as a negative number.

Syntax:

VarPer (*reference1*, *reference2*)

where *reference1* and *reference2* are references to a row, column, or cell that correspond to members of the same Account dimension whose VariancePercent results are to be calculated.

Expected Results

The following table describes the expected results when using the VariancePercent function with Accounts tagged with the below UDAs.

Table B-8 Expected Results when Using the VariancePercent Function

Col A	Col B	VarPer ([A] , [B])=0	VarPer ([A] , [B])>0	VaPer ([A] , [B])<0
Asset	Asset	0	Returns a positive value	Returns a negative value
Liability	Liability	0	Returns a negative value	Returns a positive value
Equity	Equity	0	Returns a positive value	Returns a negative value
Revenue	Revenue	0	Returns a positive value	Returns a negative value
Expense	Expense	0	Returns a negative value	Returns a positive value

VariancePercent Behavior

- The VariancePercent function expects comparison of the same account type. When you compare two account types, like Sales & Expense, the VariancePercent function performs the straight math without applying the logic of the account type. For example:

Sales	Expense	Result
-400	100	-5.

- When the VariancePercent function is applied to a dimension that is not of type Accounts, an #ERROR results at runtime.
- #MISSING is treated as zero (0), unless specified differently using ifnonnumber property.

Examples:

The VariancePercent function accepts, cell, column, or row references only.

Syntax	Example
<i>Sample syntax referencing a column</i>	VarPer ([A] , [B])
<i>Sample syntax referencing a row</i>	VarPer ([3] , [4])
<i>Sample syntax referencing a cell</i>	VarPer (Cell [3,A] , [3,B])

In this example, the VariancePercent between column A (**Actual**) and column B (**Budget**) is calculated as follows:

```
VarPer ([A] , [B])
```

This example produces the following report:

	Year Actual	Product Budget	Market VariancePercent
Sales (Income)	400,855	373,080	7%
COGS (Expense)	179,336	158,940	-13%

C

Custom-defined Functions

@HspNumToString

This function returns a string type value for the numeric value provided.

Syntax

```
@HspNumToString (NumericInput)
```

Parameter	Description
NumericInput	A numeric value that has to be converted to a string

Note

The return value is of string type.

Example

This example converts 1000 to a string so that it can be printed in a log with an @return statement.

```
@return (@hspnumtostring (1000), error);
```

1000 will be printed in the log.

@HspDateDiff

This function returns the difference between two dates in days, inclusive of dates passed.

Syntax

```
@HspDateDiff (fromDate, toDate)
```

Parameter	Description
fromDate	The From date for date comparison
toDate	The To date for date comparison

Note

The return value is an integer. The fromDate and toDate parameters can either be a date in yyyyymmdd format, an actual date value, or any variable.

Example

This example returns the number of days between 11-11-2010 and 01-01-2012, which is 461.

```
@return (@hspnumtostring (@HspDateDiff (20101111,20120101)),Error);
```

@HspDateRoll

This function returns the next date, resulting from adding noOfPeriods to startDate.

Syntax

```
@HspDateRoll (startDate, noOfPeriods, periodType)
```

Parameter	Description
startDate	The Start date from which the date has to be rolled. It must be type date.
noOfPeriods	The number of periods. If given a negative value, the date is rolled back. This must be an integer.
periodType	The Period type is an integer where: <ul style="list-style-type: none"> • 1, date roll by days • 2, date roll by months • 3, date roll by quarter • 4, date roll by half year • 5, date roll by year

Notes

The return value is a date in YYYYMMDD format.

The start date can be a date in YYYYMMDD format, a variable, or an actual date value.

Example

This example forwards the roll date by 1 month.

```
@HspDateRoll (20100101,1,2)
```

This returns 20100201.

Example

This example gets the date one quarter prior to 20101001.

```
@HspDateRoll (20100101,-1,3)
```

@HspDateToString

This function returns the string type date in DD/MM/YYYY format.

Syntax

```
@HspDateToString(NumericDateInput)
```

Parameter	Description
NumericDateInput	The date must be a numeric value.

Example

This example converts 20100101 to a string.

```
@return (@hspdatetostring (20100101), error);
```

This returns 01/01/2010.

@HspGetMbrCount

This function returns the member count of the range that is provided.

Syntax

```
@HspGetMbrCount (StringArray)
```

Parameter	Description
StringArray	An array of string values

Note

The return value is a number.

Example

This example returns the number of members in the "Element Changes" parent of the Budget Item dimension in a Oracle Hyperion Public Sector Planning and Budgeting application HCP plan type.

```
@return (@Hspnumtostring (@HspGetMbrCount (@name (@DESC ("Element Changes")))), error);
```

This returns 25 because there are 25 children under "Element Changes".

@HspGetMbrIndex

This function returns the member position in the dimension based on the outline order.

Syntax

```
@HspGetMbrIndex (String[], String)
```

Parameter	Description
String[]	An array of string values. An array in which a particular string position has to be reported.
String	The actual string in which the location in the String[] must be found.

Note

The return value is a number.

Example

This example returns the position of "2nd Element Change" under the "Element Changes" parent of the Budget Item dimension in a Oracle Hyperion Public Sector Planning and Budgeting application HCP plan type.

```
@return (@HspNumtoString (@HspMbrIndex (@Name (@DESC ("Element Changes")), @name ("2nd Element Change"))), error);
```

This returns 2.

@HspNthMbr

This function returns the nth member in the range provided.

Syntax

```
@HspNthMbr (MemberRange, PosIndex)
```

Parameter	Description
MemberRange	An array of string values
PosIndex	A number for the location

Note

The return value is a string.

Example

This example returns the 6th member under the "Element Changes" parent of the Budget Item dimension in a Oracle Hyperion Public Sector Planning and Budgeting application HCP plan type.

```
@return (@name (@HspNthMbr (@name (@DESC ("Element Changes")), 6)), error);
```

This returns "6th Element Change".

@HspStringCompare

This function compares two strings and returns "true" if both strings are the same.

Syntax

```
@HspStringCompare (SrcString, TrgtStr)
```

Parameter	Description
SrcString	The source string.
trgtStr	The target string.

Note

This function returns a boolean value.

@HspOffsetIdx

This function returns the pointer data which is the nth distinct value (where n is the offset for that data), and gets the relevant row based on the date array.

This custom function uses Oracle Hyperion Public Sector Planning and Budgeting applications for a specific purpose.

Syntax

```
@HspOffsetIdx(DataList, StartDateList, EndDateList, BaseValue, OffSet,
BaseDate)
```

Parameter
DataList
StartDateList
EndDateList
BaseValue
OffSet
BaseDate

@HspStringCompareArray

This function compares strings and returns the following numeric values (1, 2, or 3) when it meets the conditions described below:

- 1, if the source string is present in the target string
- 2, if the source string is not present in the target string
- 3, if the source string or target string are null

Syntax

```
@HspStringCompareArray (SrcString, trgtStr[])
```

Parameter	Description
SrcString	The source string.

Parameter	Description
trgtStr[]	The target string array.

Example

This example checks if "Unspecified Element" is present in descendants of the "Salary Grades" parent in the Element dimension in a Oracle Hyperion Public Sector Planning and Budgeting application HCP plan type.

```
@HspStringCompareArray(@name("Unspecified Element"),@name(@DESC("Salary Grades")));
```

This returns a 2, since "Unspecified Element" is not present in "Salary Grades".

@HspConcat

This function returns a single string by concatenating the values passed to it.

Syntax

```
@HspConcat (StringArray)
```

Parameter	Description
StringArray	An array of strings.

Note

The return value is one string.

Example

This example gets all children of "Element Changes" in log.

```
@return(@HspConcat(@name(@list("1st Element Change","","2nd Element Change"))));
```

This returns ("1st Element Change","2nd Element Change")

D

Using Smart View to Import and Edit Planning Metadata

Related Topics

- [About Using Smart View to Import and Edit Planning Metadata](#)
- [Importing Planning Dimensions in Smart View](#)
- [Editing Members in Smart View](#)
- [Adding Planning Members in Smart View](#)
- [Moving Members in Smart View](#)
- [Guidelines for Moving Members in Smart View](#)
- [Designating Shared Members in Smart View](#)
- [Refreshing and Creating Cubes in Smart View](#)

About Using Smart View to Import and Edit Planning Metadata

Administrators can use the Oracle Hyperion Planning Admin Extension with Oracle Smart View for Office to quickly import and edit dimension and member metadata.

Note:

- You must have Administrator privileges to import and edit dimension in Oracle Smart View for Office.
- All procedures described in this appendix are performed within Smart View using the Planning Admin Extension. To work with this feature, install the Planning Admin Extension. To install the Planning Admin Extension, see *Installing "EPM System Clients from EPM Workspace" in Oracle Enterprise Performance Management System Installation and Configuration Guide*.

Importing Planning Dimensions in Smart View

Importing a Oracle Hyperion Planning dimension into Oracle Smart View for Office allows you to rapidly add, edit, and move the members of the dimension.

For detailed information on the use of dimensions and members in Smart View, see "Dimensions and Members" in the *Oracle Smart View for Office User's Guide*.

- [Importing Dimensions in Smart View](#)
- [Using Smart View Grids](#)

Importing Dimensions in Smart View



Note:

You must have administrator privileges to import dimensions in Oracle Smart View for Office.

To import a Oracle Hyperion Planning dimension into a Smart View grid:

1. From the Smart View ribbon, click **Panel**.
2. From the Smart View Panel, select **Planning**.
3. Right-click **Dimensions** in a folder. The Planning dimensions in the folder are displayed.



Note:

The Time Period and Attributes dimensions are not displayed.

4. Right-click the dimension name and select **Edit Dimension**.

Using Smart View Grids

The Oracle Smart View for Office grid allows you to rapidly add, edit, and move the members of a Planning dimension.

- [Overview of the Smart View Grid](#)
- [The Smart View Grid Display](#)
- [Guidelines for Using the Smart View Grid](#)

Overview of the Smart View Grid

A Oracle Smart View for Office grid consists of two dimensions:

- A Oracle Hyperion Planning dimension on one axis
- A metadata dimension on another axis

A metadata dimension represents a flat list of metadata members. Each member in a metadata dimensions corresponds to a specific member property valid for the corresponding Planning dimension. Instead of containing numeric values, the metadata grid data cells hold the specific, corresponding property values.

The Smart View Grid Display

The Oracle Smart View for Office grid displays the Oracle Hyperion Planning dimension with member names in rows and member properties in columns.

For guidelines on performing operations in the Smart View grid, see [Guidelines for Using the Smart View Grid](#).

The ribbon displayed here is similar to the ribbon in a conventional (non-Planning) Smart View grid but contains fewer options.

For detailed information on using the Smart View ribbon, see "Ad Hoc Analysis" in the *Oracle Smart View for Office User's Guide*.

Guidelines for Using the Smart View Grid

The following guidelines will assist you in using the Oracle Smart View for Office grid to edit Oracle Hyperion Planning metadata:

- The following functionality is not available in Smart View grids with Planning metadata:
 - Pivot
 - Pivot to POV
 - Cell Text
 - Cell Notes
 - Supporting Details
- Data cell values can be textual or enumeration or numerical.
- The Parent Member is used to specify or modify the parent/child relationship.
- The position of a member in a grid does not necessarily represent the actual position of siblings in the outline.
- Each metadata grid must be linked to a corresponding Planning dimension.
- Columns for each Planning dimension are based on the corresponding set of member properties available in the Planning dimension editor.
- After a metadata grid is opened, it cannot be relinked to a different dimension.
- The corresponding valid set of metadata members is specific to each dimension.
- Planning dimension members are valid for corresponding dimensions only.

Editing Members in Smart View

The Oracle Smart View for Office grid allows you to rapidly edit the properties of members of a Oracle Hyperion Planning dimension.



Note:

Metadata grids queries are executed against the Planning business layer without interaction with Oracle Essbase.

For detailed information on the use of dimensions and members in Smart View, see "Dimensions and Members" in the *Oracle Smart View for Office User's Guide*.

To edit member properties in Smart View:

1. Within Smart View, import a Planning dimension into a Smart View grid (see [Importing Dimensions in Smart View](#)).
2. Highlight a member property in the grid.

3. In the drop-down menu, select a value.

 **Note:**

Modified cells are displayed in a different color.

4. Click **Submit Data** to save the grid.

 **Note:**

You can save changes to multiple members and properties in the same **Submit Data** operation. If Submit Data fails for one member, the Planning server will stop the operation and not save any changes.

Adding Planning Members in Smart View

The Oracle Smart View for Office grid allows you to rapidly add members to a Oracle Hyperion Planning dimension.

For detailed information on the use of dimensions and members in Smart View, see "Dimensions and Members" in the *Oracle Smart View for Office User's Guide*.

- [Selecting the Add Mode within Planning Application Properties](#)
- [Adding Members in Smart View](#)
- [Guidelines for Adding Members in Smart View](#)

Selecting the Add Mode within Planning Application Properties

There are two different modes for adding new members in the Oracle Smart View for Office grid:

- Dimension Editor Mode
- Submit without Refresh Mode

The Dimension Editor Mode requires that users perform a **Refresh** each time a member is added to a dimension but usually offers faster performance than the Submit without Refresh mode. Also, if the Dimension Editor Mode is used, new members are marked with an asterisk (*) in the grid after you perform the Refresh. The Submit without Refresh Mode does not require a Refresh but is generally slower in performance and does not mark new members.

 **Note:**

By default, new members are marked with an asterisk (*) in the grid after you perform the Refresh. You can use a different sign by adding `SMART_VIEW_DIMENSION_EDITOR_NEW_MEMBER_SUFFIX` in Oracle Hyperion Planning Application Properties.

The specific mode employed is controlled by the Smart View administrator. The mode choice is determined in Planning Application Properties.

To select the mode in Planning Application Properties by which members are added in the Smart View grid:

1. Log on to **EPM Workspace** and access Planning.
(See [Logging On to EPM Workspace and Accessing Planning](#).)
2. Access the **Planning Web UI Application Properties** page.
3. Set the value of **SMART_VIEW_DIMENSION_EDITOR_PARITY_MODE** to "False".
"False" is the default value of SMART_VIEW_DIMENSION_EDITOR_PARITY_MODE. If the value is changed to "True", the mode is Submit without Refresh.
4. Log out of Planning.
5. Log back into Planning.

 **Note:**

It is not necessary to restart the Planning server after changing the value of SMART_VIEW_DIMENSION_EDITOR_PARITY_MODE; however, you must log out and then log in again for the change to take effect.

Adding Members in Smart View

The Oracle Smart View for Office grid allows you to rapidly add members to a Oracle Hyperion Planning dimension.

For detailed information on the use of dimensions and members in Smart View, see "Dimensions and Members" in the *Oracle Smart View for Office User's Guide*.

To add members in Smart View:

1. Within Smart View, import a Planning dimension into a Smart View grid (see [Importing Dimensions in Smart View](#)).
2. Enter the name of a new member in the name column.

 **Note:**

To add a member with a numeric name, place a single quote (') in front of the numeric member name to inform the system that the number is a member name and not data. For example, to add a member named 123, enter '123.

3. Click **Refresh**.

 **Note:**

If your administrator has set the member add mode to Dimension Editor, you must perform a **Refresh** after you add a member. If the member add mode is Submit without Refresh, you are not required to perform a Refresh. If you are not certain which mode your Smart View application is using, check with your administrator.

New members are marked by an asterisk (*) in the grid after you perform the Refresh. A default set of properties is automatically applied to the new member. The default Parent Member is the root member of the dimension.

4. **Optional:** To change any property from the default value (the root member), highlight the appropriate cell in the grid, and then from the drop-down menu, select another value. (See [Editing Members in Smart View](#).)

 **Note:**

Perform a Refresh **before** you modify any properties in your new member. The Refresh operation will replace any changed values with default values from server.

5. Click **Submit Data** to save the grid.

Note the following:

- To modify the properties of the new member, see [Editing Members in Smart View](#).
- Member properties, including member names, can be localized based on the particular Locale set by the Smart View application.

Guidelines for Adding Members in Smart View

- A new member is added as the last sibling under the specified parent.
- Parent-child relationships are determined by the Parent Name column property.
- The relative position of a new member in the grid is insignificant.
- The relative position of a new member in the grid will not be changed after performing the **Submit Data** operation. To see the actual position of a new member in the outline reflected in the ad hoc grid, perform **Zoom Out**, then **Zoom In**.
- A full validity check of a new member name, including a check for invalid characters and a check for duplicate names, is performed during the **Submit Data** operation.
- Default properties are automatically applied by Oracle Hyperion Planning to a new member. The specific default properties are based upon those of the dimension.

Moving Members in Smart View

The Oracle Smart View for Office grid allows you to rapidly move members from one parent to another within a dimension.

For detailed information on the use of dimensions and members in Smart View, see "Dimensions and Members" in the *Oracle Smart View for Office User's Guide*.

To move a member in Smart View:

1. Within Smart View, import a Oracle Hyperion Planning dimension into a Smart View grid (see [Importing Dimensions in Smart View](#).)
2. Highlight the member in the **Parent Member** column in the grid.

 **Note:**

An empty Parent Member property causes the value of the Parent Member to default to the root member.

3. Enter a parent name at the intersection of the **member name** and **Parent** column.
4. Click **Submit Data** to save the grid.

 **Note:**

You can save moves of multiple members or subtrees in the same **Submit Data** operation. If the Submit Data operation fails for one member, the Planning server stops the operation and does not save changes.

Guidelines for Moving Members in Smart View

- An empty Parent Member in the grid denotes a root member in the dimension.
- The value of a Parent Member follows the rules applicable to the corresponding value in the metadata load file used by Oracle Smart View for Office.

Designating Shared Members in Smart View

The Oracle Smart View for Office grid allows you to rapidly designate members for sharing within a dimension.

 **Note:**

The Shared data storage option is not available for members of the Currency dimension in Smart View.

To designate a shared member in Smart View:

1. Verify that the base member exists.
2. Highlight the base member in the **Parent Member** column in the grid.
3. Change the parent name value for the base member.
4. Highlight the base member in the **Data Storage** column.

5. In the drop-down menu, select **Shared**.
6. Click **Submit Data** to save the grid.

The Submit Data operation will refresh the base member with its original Parent Member and Data Storage properties. The shared member will be added under the specified parent on the server.

 **Note:**

The new shared member will **not** be automatically added to the Smart View grid. The list of members displayed in the grid remains unchanged.

Refreshing and Creating Cubes in Smart View

The Oracle Smart View for Office grid allows you to quickly refresh a cube or create a new one.

To refresh or create a cube in Smart View:

1. From the Smart View ribbon, click **Panel**.
2. From the Smart View Panel, select **Planning**.
3. Right-click the folder name and select **Dimensions** in a folder. The Oracle Hyperion Planning dimensions in the folder are displayed.

 **Note:**

The Time Period and Attributes dimensions are not displayed.

4. Right-click the *root* Dimension and select **Refresh Database** or **Create Database**. The Smart View Refresh Database dialog box or Create Database dialog box is displayed.
5. Select **Refresh** or **Create**. A progress bar is displayed, indicating the percentage of steps completed for the refresh or create operation.