## Contents

### Documentation Accessibility

- Chapter 1. Using Planning
  - About Planning
  - Planning Features
  - Important Information on New Features
  - Smart View
  - Planning Web Client
  - Performance Management Architect
  - EPM Workspace
  - Lifecycle Management
  - Essbase
  - Financial Reporting
  - Business Rules
  - Web Analysis
  - Predictive Planning
  - User Licensing for Third-Party Software
  - Planning Usage Scenario
    - Deployment
    - Setting Up Applications
    - Building Plans
    - Launching Business Rules
    - Starting the Budget Review Process
    - Post-Planning Activities

### Chapter 2. Getting Started with Planning

- Overview
- Starting Essbase
- Considerations for Working with Essbase
- Understanding Implied Sharing in Planning Forms
- Connecting to Multiple Instances of Essbase
Changing Access to Forms and Folders ................................... 64
Importing Access Permissions ............................................. 65
Improving Performance When Importing Access Permissions ......... 68
Exporting Access Permissions ............................................ 68
Reporting on Access Permissions ........................................ 71
  Selecting Reporting Objects ........................................... 71
  Selecting Reporting Options .......................................... 72
  Working With Access Permissions Reports ............................ 73
Setting up Audit Trails ..................................................... 73
Examples of Tracked Actions ............................................. 74
Viewing and Clearing Audit Reports .................................... 75
Managing Security Filters .................................................. 75
Synchronizing Users With the Provision Users Utility ..................... 76
Migrating User and Group Identities ...................................... 77
Migrating Business Rule Security .......................................... 77
Removing Stale User Records ............................................ 78
Setting Up Access Permissions in Financial Reporting ..................... 79

Chapter 4. Managing Planning Databases ................................................. 81
Unlocking Applications ..................................................... 81
Using Broadcast Messaging ................................................ 82
Viewing Usage Statistics .................................................. 83
Creating and Refreshing Application Databases ............................... 83
Managing Exchange Rates ................................................ 86
  Creating Exchange Rate Tables ........................................ 86
  Editing Exchange Rate Tables ........................................... 87
  Deleting Exchange Rate Tables ......................................... 88
Managing Currency Conversions .......................................... 88
Working with Currency Conversion Calc Scripts ............................. 89
Optimizing Performance .................................................... 89
  About Reordering Dimensions ........................................ 90
  Allocating Memory for Supporting Detail Cache ....................... 90
  Configuring the Data and Index Cache Sizes ......................... 91
Other Performance Optimization Tips ................................... 91
Backing Up Applications and Application Databases .................... 91

Chapter 5. Loading Data and Metadata .................................................. 93
Overview ........................................................... 93
About Loading Dynamic Members ........................................ 93
Loading Tools ................................................................ 93
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Designing Forms with Data Validation</td>
<td>151</td>
</tr>
<tr>
<td>Designing Forms for Rolling Forecasts</td>
<td>151</td>
</tr>
<tr>
<td>Working with Forms</td>
<td>154</td>
</tr>
<tr>
<td>Selecting and Opening Forms and Folders</td>
<td>154</td>
</tr>
<tr>
<td>Previewing Forms</td>
<td>155</td>
</tr>
<tr>
<td>Printing Form Definitions</td>
<td>156</td>
</tr>
<tr>
<td>Searching for Forms</td>
<td>157</td>
</tr>
<tr>
<td>Moving Forms and Ad Hoc Grids</td>
<td>157</td>
</tr>
<tr>
<td>Deleting Forms</td>
<td>157</td>
</tr>
<tr>
<td>Renaming Forms</td>
<td>157</td>
</tr>
<tr>
<td>Importing and Exporting Forms</td>
<td>158</td>
</tr>
<tr>
<td>Specifying How Many Forms Users Can Simultaneously Use</td>
<td>159</td>
</tr>
<tr>
<td>Creating Simple Forms</td>
<td>159</td>
</tr>
<tr>
<td>Setting Form Layout</td>
<td>160</td>
</tr>
<tr>
<td>Setting Other Options</td>
<td>167</td>
</tr>
<tr>
<td>Creating Asymmetric Rows and Columns</td>
<td>168</td>
</tr>
<tr>
<td>Adding Formula Rows and Columns</td>
<td>169</td>
</tr>
<tr>
<td>Defining Simple Form Page and Point of View</td>
<td>170</td>
</tr>
<tr>
<td>Editing Forms</td>
<td>171</td>
</tr>
<tr>
<td>Creating Composite Forms</td>
<td>172</td>
</tr>
<tr>
<td>Setting Composite Form Layout</td>
<td>172</td>
</tr>
<tr>
<td>Setting Composite Form Section Properties</td>
<td>175</td>
</tr>
<tr>
<td>Setting Composite Form Point of View and Page Dimensions</td>
<td>176</td>
</tr>
<tr>
<td>Creating Master Composite Forms</td>
<td>177</td>
</tr>
<tr>
<td>Embedding Charts in Composite Forms</td>
<td>178</td>
</tr>
<tr>
<td>Using Grid Diagnostics</td>
<td>181</td>
</tr>
<tr>
<td>Using Business Rules</td>
<td>181</td>
</tr>
<tr>
<td>Selecting Business Rules</td>
<td>181</td>
</tr>
<tr>
<td>Setting Business Rule Properties</td>
<td>182</td>
</tr>
<tr>
<td>About Runtime Prompts</td>
<td>183</td>
</tr>
<tr>
<td>Understanding Runtime Prompts</td>
<td>184</td>
</tr>
<tr>
<td>About Runtime Prompts and Approvals Security</td>
<td>188</td>
</tr>
<tr>
<td>Designing Secure Runtime Prompts</td>
<td>188</td>
</tr>
<tr>
<td>Launching Business Rules With a Utility</td>
<td>189</td>
</tr>
<tr>
<td>Customizing Error Messages</td>
<td>191</td>
</tr>
<tr>
<td>Working with User Variables</td>
<td>192</td>
</tr>
<tr>
<td>About User Variables</td>
<td>192</td>
</tr>
<tr>
<td>Managing User Variables</td>
<td>192</td>
</tr>
<tr>
<td>Creating User Variables</td>
<td>193</td>
</tr>
</tbody>
</table>
Scenario 5 ................................................................. 233

Chapter 9. Managing the Budgeting Process .............................................. 237

About the Budgeting Process ................................................................. 237

Planning Units ................................................................. 237
Review Process ................................................................. 237
Planning Unit Hierarchy ................................................................. 238
Data Validation Rules ................................................................. 238
Task Lists ................................................................. 239
Setting Up Email for Approvals Notification ........................................... 239

Defining the Budgeting Process ................................................................. 239

Approvals Roles ................................................................. 239
Approvals Process ................................................................. 240
Approval Operations and Data Validations ........................................... 240

Starting and Supporting the Review Process ........................................... 241

Printing Planning Unit Annotations ........................................... 241

Managing Planning Unit Hierarchies ................................................................. 242

Creating Planning Unit Hierarchies ................................................................. 243
Assigning Planning Unit Hierarchy Scenario and Version Combinations ........................................... 248
Selecting Planning Unit Hierarchy Scenario and Version ........................................... 249
Editing Planning Unit Hierarchies ................................................................. 250
Deleting Planning Unit Hierarchies ................................................................. 250
Viewing Planning Unit Hierarchy Usage ................................................................. 251
Renaming Planning Unit Hierarchies ................................................................. 251
Synchronizing Planning Unit Hierarchies ................................................................. 252
Exporting Planning Unit Hierarchies ................................................................. 253
Importing Planning Unit Hierarchies ................................................................. 253

Planning Unit Promotional Path ................................................................. 254

Modifying the Planning Unit Promotional Path ................................................................. 254
Design Considerations for Planning Unit Promotional Path Data Validation Rules ................................................................. 257

Managing Task Lists ................................................................. 257

Creating Task List Folders ................................................................. 257
Renaming Task Lists ................................................................. 258
Renaming Task List Folders ................................................................. 258
Moving Task List Folders ................................................................. 258
Deleting Task List Folders ................................................................. 258
Creating Task Lists ................................................................. 259
Adding Instructions to Task Lists ................................................................. 259
Adding and Defining Tasks ................................................................. 259
Chapter 10. Working With an Application ........................................... 273
  Managing Forms and Folders .................................................. 273
    Creating Folders .................................................. 273
    Moving Folders .................................................. 274
    Deleting Folders .................................................. 274
    Renaming Folders ................................................. 275
  About Setting Preferences .............................................. 275
    Setting Personal Preferences ........................................ 276
    Setting Application Defaults ...................................... 276
    Specifying System Settings ....................................... 277
    Assigning Applications to Shared Services Application Groups .......... 278
  Limiting Use of an Application During Maintenance ...................... 278
  Using the MaintenanceMode Utility to Limit Application Access ........... 279
  Specifying Custom Tools ............................................ 280
  Setting Display Options ............................................. 280
  Setting Printing Options ............................................ 281
  Mapping an Application for Reporting ..................................... 281
    About the Reporting Application .................................. 281
    Creating a Reporting Application .................................. 282
    Mapping a Planning Application to a Reporting Application ............ 282
    Defining Application Mappings ..................................... 283
    Defining Dimension Mappings ....................................... 283
    Setting the Point of View ........................................ 284
    Validation Rules for Default Members ................................ 285
    Pushing Data to a Reporting Application ................................ 285
    Pushing Data with a Utility ........................................ 287
  Unsupported Features with Mapping Applications .......................... 288
  Mapping Applications and Substitution Variables .......................... 288
  Adding an Essbase Server for Reporting Applications ....................... 289
  Editing an Essbase Server for Reporting Applications ...................... 289
  Removing an Essbase Server for Reporting Applications .................... 290
Setting Up Currencies .............................................. 340
Enabling Multiple Currencies ........................................ 340
Specifying Exchange Rates ........................................... 346
Setting Up Scenarios ............................................... 347
Specifying Versions ................................................ 350
Sorting Version and Scenario Members ................................. 353
Moving Scenario and Version Members in the Dimension Hierarchy ............ 353
Adding an Aggregate Storage Outline to a Planning Application ................... 354
About Aggregate Storage ............................................ 354
Aggregate Storage Outline Plan Type Characteristics ........................ 354
Process for Adding an Aggregate Storage Database to a Planning Application ...... 355
Working with Plan Types ........................................... 355
Setting up Dynamic Time Series Members .................................. 356
Additional Supported Planning Application Features .......................... 358
Considerations for Alternate Hierarchies in Period Dimensions ................ 358
Working with Applications that Use Planning Application Administration ........... 359
Refreshing Application Databases Using a Utility .......................... 359
Refreshing Databases with Level 0 Members Set to Dynamic Calc .................. 360
Working With Essbase Partitions ...................................... 360
Sorting Members Using a Utility ...................................... 361
Deleting Shared Descendants Using a Utility ................................ 362
Using Smart Lists, UDAs, and Member Formulas ............................... 363
Working with Smart Lists ........................................... 363
Working with UDAs ............................................... 366
Working with Member Formulas ...................................... 368

Chapter 13. Customizing Planning Web Client ............................................. 375
Customizing Reports ................................................................ 375
Customizing the Planning Tools Page ...................................... 377
Customizing Cascading Style Sheets ...................................... 378
Location of Cascading Style Sheet Files ..................................... 379
Modifying Cascading Style Sheets ....................................... 379
Customization Examples ............................................ 380
Customizing the Style of Row and Column Members in Forms ................. 381
Customizing the Style of Header Cells in Forms .............................. 382
Applying Bolding to All Forms ........................................ 382
Applying Bolding to One Form ....................................... 382
Customizing Skins to Add Components .................................... 382
Registering Customized Skins ........................................... 383
Appendix A. Customizing Forms with JavaScript .......................................................... 409
About Customizing Forms .................................................................................. 409
Modifying JavaScript ..................................................................................... 409
customCellEnterPre ................................................................................... 410
customCellEnterPost ................................................................................. 410
customCellValidatePre ............................................................................. 411
customCellValidatePost ............................................................................ 412
customOnLoad .......................................................................................... 413
drawCustomButtons ............................................................................... 413
validateForm ............................................................................................. 413
Deploying Custom JavaScript ........................................................................ 414

Appendix B. Naming Restrictions ........................................................................ 415
Restrictions for Applications and Databases .................................................. 415
Restrictions for Dimensions, Members, Aliases, and Forms ......................... 416
Dimension and Member Names in Calc Scripts, Report Scripts, Formulas, Filters, and Substitution Variables ................................................. 417
Restrictions for User Names .......................................................................... 418

Appendix C. Form Formula Functions ............................................................... 419
Using Formulas and Formula Functions ....................................................... 419
Creating Formulas ...................................................................................... 419
Editing Formulas ......................................................................................... 419
Deleting Formulas ....................................................................................... 420
Formula Functions ......................................................................................... 420
Arguments ..................................................................................................... 421
Abs .............................................................................................................. 425
Average ......................................................................................................... 426
AverageA ...................................................................................................... 426
Count ............................................................................................................ 427
CountA .......................................................................................................... 428
Difference .................................................................................................... 428
Eval ............................................................................................................... 429
IfThen, If ...................................................................................................... 430
Max .............................................................................................................. 433
Min .............................................................................................................. 433
Mod .............................................................................................................. 434
PercentOfTotal .......................................................................................... 434
Pi ............................................................................................................... 435
Product ....................................................................................................... 435
Documentation Accessibility

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

Access to Oracle Support

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

Follow EPM Information Development on these social media sites:
LinkedIn - http://www.linkedin.com/groups?gid=3127051&goback=.gmp_3127051
Twitter - http://twitter.com/hyperionepminfo
Facebook - http://www.facebook.com/pages/Hyperion-EPM-Info/102682103112642
Google+ - https://plus.google.com/106915048672979407731/#106915048672979407731/posts
YouTube - http://www.youtube.com/user/OracleEPMWebcasts
About Planning

Oracle Hyperion Planning is a budgeting and planning solution that drives collaborative, event-based operational planning processes throughout the organization for a wide range of financial and operational needs. It enables a complete and closed-loop planning process that drives continuous business improvement. Decision makers and managers can communicate the course of action and collaborate with budget holders to optimize the planning process. Planners have the flexibility to adapt rapidly, ensuring plans are relevant and useful.

Planning Features

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
• Lowers 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 Planning new features, see Oracle Hyperion Planning New Features for the current release.

Smart View
Smart View is the Microsoft Office interface to Planning, where users can perform their planning and forecasting activities. Smart View also enables users to work offline in disconnect mode. 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
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. Other administrative features, such as creating and administering applications and dimensions, are now available through Oracle Hyperion EPM Architect.

Performance Management Architect
Performance Management Architect is a component of Planning installation and configuration. You use it to create and work with Planning applications and dimensions, Smart Lists, UDAs, member formulas, and other features. You can also create business rules.

With Performance Management Architect, you can view, create, and validate Performance Management Architect applications, and deploy them to create Planning applications. Deploying applications from Performance Management Architect to Planning is a long-running operation. The initial deployment may take more time than subsequent re-deployments.
Performance Management Architect applications can contain business rules created using the Oracle Hyperion Calculation Manager module in Performance Management Architect. For assistance on tasks performed in Performance Management Architect, see the Oracle Hyperion Enterprise Performance Management Architect Administrator’s Guide. For information on installing and configuring Performance Management Architect, see the Oracle Enterprise Performance Management System Installation and Configuration Guide.

EPM Workspace

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” on page 48.

Lifecycle Management

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

Lifecycle Management features:

- Viewing applications and folders
- Searching for artifacts
- 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**

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” on page 303. For information on data sources for Planning applications created in Performance Management Architect, see the *Oracle Hyperion Enterprise Performance Management Architect Administrator’s Guide*.

**Financial Reporting**

Oracle Hyperion Financial Reporting is a management reporting tool that transforms data into meaningful business information through highly formatted reports. 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**


**Web Analysis**

Oracle Hyperion Web Analysis is an analysis, presentation, and reporting solution. It allows organizations to deliver information to large user communities at a low cost.

**Predictive Planning**

With Predictive Planning, users can work in 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 Smart View, users can access Predictive Planning functionality.
from the Planning menu or ribbon. (For installation instructions, see the Oracle Enterprise Performance Management System Installation and Configuration Guide.)

User Licensing for Third-Party Software

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

Planning Usage Scenario

You create applications using Performance Management Architect application administration, and deploy them to create Planning applications. Applications contain dimensions and dimension attributes designed to meet Planning needs, such as accounts, entities, scenarios, and other dimension elements. You can create an unlimited number of applications. For information about Performance Management Architect, see the Oracle Hyperion Enterprise Performance Management Architect Administrator’s Guide.

These topics outline how Planning is used:

- “Deployment” on page 25
- “Setting Up Applications” on page 26
- “Building Plans” on page 31
- “Launching Business Rules” on page 31
- “Starting the Budget Review Process” on page 31
- “Post-Planning Activities” on page 33

Deployment

An IT professional installs and configures Planning on server computers and client workstations. The IT professional should have experience administering relational databases, installing client/server and Web server software, and configuring user directories.

<table>
<thead>
<tr>
<th>User</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>IT professional</td>
<td>Installs Planning system requirements, including operating systems, relational database, Web server, application server, and Web browser</td>
</tr>
<tr>
<td>IT professional</td>
<td>Installs Essbase</td>
</tr>
<tr>
<td>IT professional</td>
<td>Sets up the data source name (DSN) for the OLAP and relational databases</td>
</tr>
<tr>
<td>IT professional</td>
<td>Installs Planning and Financial Reporting</td>
</tr>
<tr>
<td>IT professional</td>
<td>Creates relational databases and connections</td>
</tr>
<tr>
<td>User</td>
<td>Task</td>
</tr>
<tr>
<td>-----------------</td>
<td>--------------------------------------------------------</td>
</tr>
<tr>
<td>IT professional</td>
<td>Configures the Web server</td>
</tr>
<tr>
<td>IT professional</td>
<td>Sets up users in an external user directory</td>
</tr>
<tr>
<td>IT professional</td>
<td>Optional: Creates a test application with users</td>
</tr>
<tr>
<td>IT professional</td>
<td>Tests installation and configuration</td>
</tr>
</tbody>
</table>

**Table 2  Client Installation Users and Tasks**

<table>
<thead>
<tr>
<th>User</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>IT professional</td>
<td>Installs the Planning remote Windows client, Smart View, Microsoft Excel, and Financial Reporting</td>
</tr>
<tr>
<td>IT professional</td>
<td>Installs a Web browser</td>
</tr>
<tr>
<td>IT professional</td>
<td>Tests connections to the Web and network</td>
</tr>
</tbody>
</table>

**Setting Up Applications**

**Subtopics**

- Creating Metadata
- Defining Users and Access Permissions
- Creating and Refreshing Applications
- Designing Forms
- Designing Worksheets
- Populating Applications with Data
- Creating Business Rules
- Setting Targets
- Creating Tasks Lists
- Reporting
- Initializing Planning Cycles

Setting up applications includes the tasks listed in this section. Administrators can also set up applications using Performance Management Architect application administration, and deploy them to Planning. See the *Oracle Hyperion Enterprise Performance Management Architect Administrator’s Guide*.

**Creating Metadata**

Consulting partners and the Planning administrator design applications and create an application framework with Performance Management Architect application administration. There is only one application owner per application. The owner can grant ownership to another administrator. The application framework includes:

- Application name
- Number and names of plan types
- Calendar structure
- Whether the application is a multicurrency application
- Default currency

After the application framework is created, dimensions and members are added to the application and assigned to the plan type in Performance Management Architect. There are up to eight required dimensions in an application:
- Currency (for multicurrency applications)
- HSP_Rates (for multicurrency applications)
- Account
- Entity
- Scenario
- Version
- Period
- Year

Table 3  Creating Metadata

<table>
<thead>
<tr>
<th>User</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administrator</td>
<td>Designs and creates applications</td>
</tr>
<tr>
<td>Administrator</td>
<td>Defines metadata (currencies, scenarios, versions, calendar)</td>
</tr>
<tr>
<td>Administrator</td>
<td>Enters exchange rates</td>
</tr>
<tr>
<td>Administrator</td>
<td>Defines attributes and attribute values</td>
</tr>
<tr>
<td>Administrator</td>
<td>Loads custom dimensions</td>
</tr>
</tbody>
</table>

### Defining Users and Access Permissions

An IT professional defines users and groups and configures a user directory before users can access Planning applications. For detailed information, see the Oracle Enterprise Performance Management System User Security Administration Guide. Within Planning, you assign users or groups to secured objects.

Table 4  Setting Up Users and Access Permissions

<table>
<thead>
<tr>
<th>User</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>IT professional</td>
<td>Sets up Planning users with a supported user directory</td>
</tr>
<tr>
<td>Administrator</td>
<td>Synchronizes users in the user directory with a Planning application</td>
</tr>
<tr>
<td>Administrator</td>
<td>Assigns access permissions to users and groups</td>
</tr>
</tbody>
</table>
Creating and Refreshing Applications

Administrators create and periodically refresh the Essbase outline and security structure for the Planning application. The outline and security structure are created based on metadata stored in the Planning application’s relational database. Planning creates:

- An Essbase application
- Multiple Essbase databases (one per plan type)
- Essbase access permissions filters
- Essbase outlines (all metadata):
  - Members
  - Shared members
  - User-defined attributes and attribute values
  - Exchange rates

<table>
<thead>
<tr>
<th>User</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administrator</td>
<td>Assigns access permissions to dimensions, members, forms, folders, and task lists in Planning (see Chapter 3, “Setting Up Access Permissions”)</td>
</tr>
</tbody>
</table>

Table 5  Generating Databases

Designing Forms

Table 6  Designing Forms

<table>
<thead>
<tr>
<th>User</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administrator and interactive user</td>
<td>Creates and maintains forms in Planning</td>
</tr>
<tr>
<td>Administrator and interactive user</td>
<td>Creates and manages folders in Planning for form management</td>
</tr>
<tr>
<td>Administrator</td>
<td>Assigns access permissions to forms and folders</td>
</tr>
</tbody>
</table>

Designing Worksheets

Planning users can install Smart View to work with forms in Excel to leverage Excel worksheet models, build custom formulas, format reports, and work disconnected from Planning.
Populating Applications with Data

Administrators and interactive users use Performance Management Architect application administration to pre-populate Planning applications with data. See the *Oracle Hyperion Enterprise Performance Management Architect Administrator’s Guide*.

<table>
<thead>
<tr>
<th>Table 7</th>
<th>Populating Applications with Data</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>User</strong></td>
<td><strong>Task</strong></td>
</tr>
<tr>
<td>Administrator</td>
<td>Use Copy Data to copy plans from one dimensional intersection to another, including relational data and supporting detail</td>
</tr>
<tr>
<td>Administrator and interactive user</td>
<td>Performs bulk loads</td>
</tr>
<tr>
<td>Any user</td>
<td>Enters values for global saved assumptions referenced in business rules</td>
</tr>
<tr>
<td>Any user</td>
<td>Enters data into the Planning application through forms or Smart View</td>
</tr>
<tr>
<td>Any user</td>
<td>Calculates and creates data using business rules or Copy Version</td>
</tr>
</tbody>
</table>

Creating Business Rules

<table>
<thead>
<tr>
<th>Table 8</th>
<th>Creating Business Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>User</strong></td>
<td><strong>Task</strong></td>
</tr>
<tr>
<td>Administrator</td>
<td>Assigns users access permissions to business rules</td>
</tr>
<tr>
<td>Administrator and interactive user</td>
<td>Creates business rules. See the <em>Oracle Enterprise Performance Management System User Security Administration Guide</em> for information on Calculation Manager roles and rights.</td>
</tr>
<tr>
<td>Administrator, interactive user, and planners who are assigned launch access permissions by an administrator</td>
<td>Launches business rules for Planning applications</td>
</tr>
</tbody>
</table>

For information on administering business rules, see the *Oracle Hyperion Enterprise Performance Management Architect Administrator’s Guide*.

See also “Using Business Rules” on page 181.

Setting Targets

Administrators set target type versions for Planning applications. User access to target data is determined by access permissions. Typically, target data is stored in the upper levels of metadata, such as Business Unit. Administrators configure the Essbase database so target data is not replaced by lower-level #MISSING values.

Target data is typically used as guidance for data entry and analysis. Create it, for example, though:

- Manual data entry
Modeling with business rules
Pushing targets down to lower-level members using business rules

Table 9  Setting Targets

<table>
<thead>
<tr>
<th>User Type</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administrator</td>
<td>Adds a standard target-type version</td>
</tr>
<tr>
<td>Administrator</td>
<td>Assigns users access permissions (typically, read-only) to the target version</td>
</tr>
<tr>
<td>Administrator</td>
<td>Configures the Essbase database so target data is not replaced by lower-level #MISSING values</td>
</tr>
<tr>
<td>Any user</td>
<td>Creates target data</td>
</tr>
<tr>
<td>Administrator</td>
<td>Publishes targets using Financial Reporting</td>
</tr>
<tr>
<td>Any user</td>
<td>Displays targets on forms for guidance or input</td>
</tr>
</tbody>
</table>

Creating Tasks Lists

Table 10  Task Lists

<table>
<thead>
<tr>
<th>User Type</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administrator</td>
<td>Creates task lists that structure the budgeting process for users</td>
</tr>
<tr>
<td>Administrator</td>
<td>Assigns access permissions to task lists and tasks</td>
</tr>
</tbody>
</table>

Reporting

Use Financial Reporting to create reports for Windows or Web-enabled clients. Financial Reporting uses Essbase databases and adheres to Essbase access permissions filters generated by Planning.

Table 11  Reporting

<table>
<thead>
<tr>
<th>User Type</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administrator and interactive user</td>
<td>Creates and launches reports using Financial Reporting</td>
</tr>
<tr>
<td>Any user</td>
<td>Prints reports</td>
</tr>
<tr>
<td>Any user</td>
<td>Views reports throughout the planning cycle</td>
</tr>
</tbody>
</table>
Initializing Planning Cycles

Table 12  Initializing Planning Cycles

<table>
<thead>
<tr>
<th>User Type</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administrator</td>
<td>Selects planning units for iterative review, analysis, and approval</td>
</tr>
<tr>
<td>Any user</td>
<td>Specifies whether to receive email after becoming planning unit owners</td>
</tr>
</tbody>
</table>

Building Plans

Administrators define task lists to guide users through the planning process. Planning users start the planning cycle by logging into an application and opening forms. Users read form instructions and review historical or target data for guidance when preparing plans. For additional user tasks, see the Oracle Hyperion Planning User’s Guide.

Users can save data in forms, and can refresh currently stored data values in forms before saving. Users can restore data to the last saved version, and save a personal version using Copy Version. When users save:

- Data is saved to the Essbase database.
- Subtotals on forms are calculated and stored if the Calculate Form business rule is selected to run on save when forms are created. (By default, this business rule is not selected to run on save.)
- Business rules are launched to calculate data.

Launching Business Rules

The business rules available to users depend on access permissions. Administrators can assign launch access permissions to business rules from within Planning (see Chapter 3, “Setting Up Access Permissions”). For more information on business rule access permissions, also see the Oracle Enterprise Performance Management System User Security Administration Guide.

Users with launch permissions can launch business rules from Planning or Smart View.

Starting the Budget Review Process

The administrator initializes review processes by placing planning units in the First Pass state. Data is typically promoted for review after users enter data, calculate, and save. When users promote planning units, they enter annotations that record assumptions and select the next owners or reviewers. Promoting planning units:

- Assigns ownership to the selected user
- Overrides access permissions on the planning unit to read-only for non-owners
- Changes planning unit status from First Pass to Under Review
- Updates status for upper-level (or lower-level) planning units to Under Review as necessary
Records an audit entry in the process history
Optionally sends email notifications to the application owner and new planning unit owner

Reviewers can make modifications and annotations. Read-only users can enter planning unit annotations and account annotations, but cannot enter data values. Users can keep a personal copy of the original data to track modified data. When planning units are promoted, the reviewer is the new owner and typically:

- Receives email notifications and is directed to the Process Definition page
- Sorts the status page by entity, process state, or current owner
- Reviews annotations and planning unit history
- Opens forms and reviews, analyzes, modifies, and calculates data
- Makes annotations (including account annotations)
- Promotes or rejects the planning unit, with iterations until the planning unit is approved

After planning units are approved, the application owner becomes the planning unit owner. Only administrators can reject planning units. To close the review cycle, the application owner or administrator changes the scenario or version to read-only for all users.

<table>
<thead>
<tr>
<th>Table 13</th>
<th>Starting the Budget Review Process</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Users</strong></td>
<td><strong>Tasks</strong></td>
</tr>
<tr>
<td>Administrators</td>
<td>Starts the planning unit for the planning cycle</td>
</tr>
<tr>
<td>Users with access</td>
<td>Enters data</td>
</tr>
<tr>
<td>Users with access</td>
<td>Calculates data</td>
</tr>
<tr>
<td>Users with access</td>
<td>Optionally, creates a copy of submissions</td>
</tr>
<tr>
<td>Users with access</td>
<td>Promotes data for review</td>
</tr>
<tr>
<td>Users with access</td>
<td>Enters annotations to support the review process. Until an owner is established for a planning unit, multiple users can update it. Access permissions for non-owners are changed to read-only when the planning unit is promoted to the next reviewer.</td>
</tr>
<tr>
<td>Users with access</td>
<td>Notifies reviewers that the planning unit is ready for review</td>
</tr>
<tr>
<td>Users with access</td>
<td>Reviews data in the promoted planning unit</td>
</tr>
<tr>
<td>Users with access</td>
<td>Reads and modifies promoted data through forms</td>
</tr>
<tr>
<td>Users with access</td>
<td>Runs business rules</td>
</tr>
<tr>
<td>Users with access</td>
<td>Reads promoted data using Financial Reporting, Oracle Hyperion Web Analysis, Smart View, or third-party reporting tools</td>
</tr>
<tr>
<td>Users with access</td>
<td>Modifies data values, enters annotations, signs off on planning units, and promotes, rejects, and approves planning units</td>
</tr>
<tr>
<td>Administrators</td>
<td>Checks the planning process status</td>
</tr>
<tr>
<td>Users</td>
<td>Tasks</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>Administrators</td>
<td>Reads supporting annotations</td>
</tr>
<tr>
<td>Administrators</td>
<td>Views the audit trail</td>
</tr>
<tr>
<td>Administrators</td>
<td>Completes the planning cycle</td>
</tr>
<tr>
<td>Administrators</td>
<td>Changes user access of scenarios and versions in Essbase to read-only</td>
</tr>
<tr>
<td>Administrators and interactive users</td>
<td>Publishes reports using Financial Reporting</td>
</tr>
</tbody>
</table>

### Post-Planning Activities

**Table 14  Post-Planning Activities**

<table>
<thead>
<tr>
<th>Users</th>
<th>Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administrators and interactive users</td>
<td>Upload planning data by scenario to other applications, a general ledger, or an ERP</td>
</tr>
<tr>
<td>Users with access</td>
<td>Copies a version of the completed plan to a new version or copies data from one dimensional intersection to another</td>
</tr>
<tr>
<td>Administrator and interactive user</td>
<td>Performs bulk loads of the latest actuals data to report on actual performance versus plan</td>
</tr>
<tr>
<td>Users with access</td>
<td>Accesses forms, Financial Reporting, Smart View, Oracle Hyperion Performance Scorecard, or third-party reporting tools to read updates to actuals and analyze performance to plan</td>
</tr>
</tbody>
</table>
Overview

A 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.

You can work with applications using Performance Management Architect and Planning application administration. See the Oracle Hyperion Enterprise Performance Management Architect Administrator’s Guide for these Performance Management Architect application administration tasks:

- Create an application in Performance Management Architect and deploy it to create a Planning application.
- Create entities to reflect your organization’s budget structure.
- Create accounts and dimensions for gathering planning data.
- Create scenarios and versions.
- Modify the calendar with any custom summary time periods needed by your organization.
- For multicurrency applications, specify the base currency, currency code and symbol, triangulation currency, reporting currency, and exchange rate type.
Create business rules in Calculation Manager.

See this guide for Planning application administration tasks, including:

- Create, delete, and register Planning applications.
- Create forms, task lists, and menus.
- Manage currency conversion and exchange rate tables.
- Identify the review and approval process, requirements, and participants, and manage the budgeting process.

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

Starting Essbase

The data for Planning applications resides on the 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 Essbase 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 “Creating and Refreshing Application Databases” on page 83.

If you are not using Performance Management Architect, you work with applications using Planning application administration. If you upgrade an application created in Planning application administration to Performance Management Architect, you cannot return to working with that application in Planning application administration. Modifying Essbase outlines directly through Oracle Essbase Administration Services is not supported.

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 calc. Other UDAs, such as custom UDAs and the **HSP_NOLINK** UDA, are retained only if defined in Performance Management Architect or in Planning (for Planning application administration). To create and update UDAs, see “Working with UDAs” on page 366.

Because member formula support is available in Performance Management Architect and Planning (for Planning application administration), there is less need for the **HSP_UDF** UDA. Every effort should be made to define UDAs in Performance Management Architect or in Planning (for 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. Performance Management Architect and Planning support 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” on page 91.

**Understanding Implied Sharing in Planning Forms**

With 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. 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” on page 398.

### Connecting to Multiple Instances of Essbase

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

- When creating a Planning application with the Planning Application Wizard or in Performance Management Architect (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 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 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:
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 15  Examples of Configurable Properties

<table>
<thead>
<tr>
<th>PROPERTY_NAME Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RDB_DRIVER</td>
<td>See “Changing the JDBC Driver” on page 40.</td>
</tr>
<tr>
<td>RDB_SERVER_URL</td>
<td></td>
</tr>
<tr>
<td>JDBC_MIN_CONNECTIONS</td>
<td>See “Configuring JDBC Connection Pooling” on page 41.</td>
</tr>
<tr>
<td>JDBC_MAX_CONNECTIONS</td>
<td></td>
</tr>
<tr>
<td>OFFLINE_COMPRESSION_THRESHOLD</td>
<td>See “Setting Offline Compression” on page 42.</td>
</tr>
<tr>
<td>SUPPORTING_DETAIL_CACHE_SIZE</td>
<td>See “Allocating Memory for Supporting Detail Cache” on page 90.</td>
</tr>
<tr>
<td>SUBST_VAR_CACHE_LIFETIME</td>
<td>See “About Selecting Substitution Variables as Members” on page 208.</td>
</tr>
<tr>
<td>DIRECT_DATA_LOAD</td>
<td>See “Loading Data” on page 101.</td>
</tr>
<tr>
<td>DATA_LOAD_FILE_PATH</td>
<td></td>
</tr>
<tr>
<td>OLAP_PATH_SEPARATOR</td>
<td>See “Configuring Data Synchronization on UNIX” on page 42.</td>
</tr>
<tr>
<td>OLAP_MAX_CONNECTIONS</td>
<td>See “Optimizing Performance” on page 89.</td>
</tr>
<tr>
<td>ENABLE_FOR_OFFLINE</td>
<td>See “Setting Other Options” on page 167.</td>
</tr>
<tr>
<td>SMART_VIEW_DISPLAY_WARNING</td>
<td>See “Controlling Smart View Messages” on page 42.</td>
</tr>
<tr>
<td>SMART_VIEW_MD_NEW_MEMBER_SUFFIX</td>
<td>See “Selecting the Add Mode within Planning Application Properties” on page 448</td>
</tr>
<tr>
<td>HOME_PAGE</td>
<td>See “Setting the Application Home Page” on page 43.</td>
</tr>
<tr>
<td>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</td>
<td>See “Setting Background Processing” on page 44.</td>
</tr>
<tr>
<td>BUSINESS_RULE_LAUNCH_LOG_RTP_VALUES</td>
<td></td>
</tr>
<tr>
<td>BUSINESS_RULE_LAUNCH_LOG_CALC_SCRIPT</td>
<td></td>
</tr>
<tr>
<td>CAPTURE_RTP_ON_JOB_CONSOLE</td>
<td>See “About Runtime Prompts” on page 183.</td>
</tr>
<tr>
<td>PROPERTY_NAME Column</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MAX_VALIDATION_RECORDS</td>
<td>See “Setting Data Validation Properties” on page 46.</td>
</tr>
<tr>
<td>VALIDATION_CACHE_SIZE</td>
<td></td>
</tr>
<tr>
<td>MAX_CELL_TEXT, MAX_CELL_NOTE</td>
<td>See “Setting the Maximum Length for Text Values and Comments in Cells” on page 47.</td>
</tr>
<tr>
<td>EDIT_DIM_ENABLED</td>
<td>See “Managing Dimensions from Within Deployed Planning Applications” on page 47.</td>
</tr>
<tr>
<td>ORACLE_ADF_UI</td>
<td>See “Using the Planning Release 11.1.2.1 User Interface and Features” on page 48.</td>
</tr>
<tr>
<td>DATA_GRID_CACHE_SIZE</td>
<td>See “Specifying How Many Forms Users Can Simultaneously Use” on page 159.</td>
</tr>
</tbody>
</table>

**Note:** In some cases, additional properties may be visible in the Properties page. Because the following properties are required by 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 Oracle Hyperion Enterprise Performance Management System Configurator.

**Changing the JDBC Driver**

By default, Planning uses the embedded Oracle JDBC driver. To change the JDBC driver, update the JDBC property using these values:
### Table 16  JDBC Property Values

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

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 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

Configuring Data Synchronization on UNIX

When you work with Planning applications in Performance Management Architect, running integrations in Data Synchronization fails if Essbase is installed on a UNIX computer. To resolve this issue, in Planning, select Administration, then Manage Properties. Click Application Properties and add the value OLAP_PATH_SEPARATOR=/ (The default value is \\n, which is for the Windows environment.) Click Save and restart the Planning server.

Setting Offline Compression

You can set up compression for applications taken offline by adding the OFFLINE_COMPRESSION_THRESHOLD property and setting the threshold, in bytes, for when to start using compression. Compression is enabled when the server’s response to offline client requests is greater than the Offline Compression Threshold number.

➤ To set offline compression:
1 Select Administration, then Application, and then Properties.
2 Select System Properties to set properties for all Planning applications.
3 Update the Offline Compression Threshold setting:
   • To add the property, click Add. In the blank row, enter OFFLINE_COMPRESSION_THRESHOLD (avoid using spaces). Under Property Value, enter the number of bytes to use for the threshold for when to compress offline applications.
   • To change the property, change its value under Property Value. To disable compression, enter 0.
   • To delete the property, select its name, and press Delete.
4 Click Save and confirm your changes.
5 Stop and then restart the application server.

Controlling Smart View Messages

If you use different releases of Smart View and 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 Planning metadata in Smart View. (See “Importing Planning Dimensions in Smart View” on page 445.) New members created in the Smart View grid are, by default, marked with an asterisk or star(*) 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 or star(*) 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 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 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 17  Job Property Names and Values

<table>
<thead>
<tr>
<th>PROPERTY_NAME Column</th>
<th>Property Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RULE_MAX_WAIT</td>
<td>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.</td>
<td>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. Note: Business rules that are set to automatically run when a form is loaded or saved never run in the background.</td>
</tr>
<tr>
<td>RULE_MONITOR_DELAY</td>
<td>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.</td>
<td>The interval in milliseconds for checking the status of business rules.</td>
</tr>
<tr>
<td>CLR_CELL_MAX_WAIT</td>
<td>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.</td>
<td>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.</td>
</tr>
<tr>
<td>CLR_CELL_MONITOR_DELAY</td>
<td>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.</td>
<td>The interval in milliseconds for checking the status of Clear Cell Detail operations.</td>
</tr>
<tr>
<td>COPY_DATA_MAX_WAIT</td>
<td>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.</td>
<td>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.</td>
</tr>
<tr>
<td>COPY_DATA_MONITOR_DELAY</td>
<td>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.</td>
<td>The interval in milliseconds for checking the status of Copy Data jobs.</td>
</tr>
<tr>
<td>PROPERTY_NAME</td>
<td>Property Value</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-------------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>PUSH_DATA_MONITOR_DELAY</td>
<td>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.</td>
<td>The interval in milliseconds for checking the status of Push Data jobs.</td>
</tr>
<tr>
<td>PUSH_DATA_MAX_WAIT</td>
<td>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.</td>
<td>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.</td>
</tr>
<tr>
<td>JOB_STATUS_MAX_AGE</td>
<td>Specify the value in milliseconds. The default value is 4 days (set in milliseconds). There is no minimum value.</td>
<td>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. 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. Removing completed job records can improve performance.</td>
</tr>
</tbody>
</table>

4 Click Save and confirm your changes.

### Setting Business Rules Launch Logging Properties

You can set application properties to enable logging when 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 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.
Select Application Properties to set properties and values for a Planning application.

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

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 Chapter 8, “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 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 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.

Managing Dimensions from Within Deployed Planning Applications

By default, administrators manage dimensions and members in Performance Management Architect. However, adding and enabling the EDIT_DIM_ENABLED property enables
administrators to manage dimensions only from deployed Planning applications, not from within Performance Management Architect.

To use the \texttt{EDIT\_DIM\_ENABLED} property, add it using the instructions in “Setting Application and System Properties” on page 39 and set its value to \texttt{true}.

\begin{caution}
If you add and set this property, change dimensions and members, and then redeploy the application from Performance Management Architect, all your changes will be overwritten by the content deployed from Performance Management Architect.
\end{caution}

\section*{Using the Planning Release 11.1.2.1 User Interface and Features}

After upgrading to the current release, if you prefer the user interface and features from Planning Release 11.1.2.1, you can add the \texttt{ORACLE\_ADF\_UI} application property to use that release’s interface and functionality. To revert to the Planning release 11.1.2.1 user interface and features, add the application property \texttt{ORACLE\_ADF\_UI} (described in “Setting Application and System Properties” on page 39), and set its value to \texttt{false}. To revert to the user interface and features for the current release, change the property setting to \texttt{true} or delete the property.

\begin{note}
Keep in mind that new features for the current release are available only if \texttt{ORACLE\_ADF\_UI} is set to \texttt{true}. Setting \texttt{ORACLE\_ADF\_UI} to \texttt{false} enables features from Planning Release 11.1.2.1.
\end{note}

You can access the documentation for Planning Release 11.1.2.1 here:

http://docs.oracle.com/cd/E17236_01/index.htm

\begin{caution}
Oracle recommends that, if you plan to use this property, you set it before you make changes in your Planning application; if you work with the new functionality in the current release and then revert to the earlier interface, you may lose your changes. Oracle also recommends against repeatedly changing this property.
\end{caution}

\section*{Logging On to EPM Workspace and Accessing Planning}

You work with Planning in the EPM Workspace environment. The default EPM Workspace URL is \texttt{http://web server:port/workspace/}, where \texttt{web server} is the Web server machine hostname and \texttt{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 \textit{Oracle Enterprise Performance Management System Installation and Configuration Guide}. 

48  Getting Started with Planning
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.

For information on Performance Management Architect application administration tasks, see the Oracle Hyperion Enterprise Performance Management Architect Administrator'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

#### Subtopics

- Suppressing Password Prompts in Planning Utilities
- About Text Files for Planning Utilities
- Running Planning Utilities on UNIX
- About EPM Oracle Instance

Administrators can use the following utilities with 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>
<thead>
<tr>
<th>Utility</th>
<th>Description</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>BroadcastMessage</td>
<td>Use broadcast messaging to communicate a text message to all Planning users currently logged on to an application.</td>
<td>“Using Broadcast Messaging” on page 82</td>
</tr>
<tr>
<td>CalcMgrCmdLineLauncher</td>
<td>Launch business rules or rulesets created with Calculation Manager.</td>
<td>“Launching Business Rules With a Utility” on page 189</td>
</tr>
<tr>
<td>CubeRefresh</td>
<td>Create and refresh application databases used to store data in Essbase for each plan type in the application.</td>
<td>“Working with Applications that Use Planning Application Administration” on page 359</td>
</tr>
<tr>
<td>DeleteSharedDescendant</td>
<td>Delete shared dimension members that are descendants of a given member.</td>
<td>“Working with Applications that Use Planning Application Administration” on page 359</td>
</tr>
<tr>
<td>ExportSecurity</td>
<td>Export Planning access permissions to a file, enabling you to export and import access permissions across applications.</td>
<td>“Exporting Access Permissions” on page 68</td>
</tr>
<tr>
<td>FormDefUtil</td>
<td>Move form definitions between Planning applications by exporting or importing to or from an XML file.</td>
<td>“Importing and Exporting Forms” on page 158</td>
</tr>
<tr>
<td>HspUnlockApp</td>
<td>Clear all records in the HSP_LOCK table.</td>
<td>“Unlocking Applications” on page 81</td>
</tr>
<tr>
<td>Utility</td>
<td>Description</td>
<td>See</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------------------------------------------------------</td>
</tr>
<tr>
<td>ImportFormDefinition</td>
<td>Import a form definition from a text file into a Planning form.</td>
<td>&quot;Importing Form Definitions&quot; on page 194</td>
</tr>
<tr>
<td>ImportSecurity</td>
<td>Load access permissions for users or groups from a text file into Planning.</td>
<td>&quot;Importing Access Permissions&quot; on page 65</td>
</tr>
<tr>
<td>MaintenanceMode</td>
<td>Grant and withdraw access to Planning applications during maintenance.</td>
<td>&quot;Using the MaintenanceMode Utility to Limit Application Access&quot; on page 279</td>
</tr>
<tr>
<td>Outline Load</td>
<td>Load metadata and data for Planning applications.</td>
<td>&quot;Working with the Outline Load Utility&quot; on page 95</td>
</tr>
<tr>
<td>PasswordEncryption</td>
<td>Enable suppressing password prompts when running Planning utilities that prompt for passwords.</td>
<td>&quot;Suppressing Password Prompts in Planning Utilities&quot; on page 52</td>
</tr>
<tr>
<td>ProvisionUsers</td>
<td>Synchronize Planning users, groups, and roles in the Shared Services Console with a Planning application and with Essbase.</td>
<td>&quot;Synchronizing Users With the Provision Users Utility&quot; on page 76</td>
</tr>
<tr>
<td>PushData</td>
<td>Schedule pushing data to a reporting application.</td>
<td>&quot;Pushing Data with a Utility&quot; on page 287</td>
</tr>
<tr>
<td>SortMember</td>
<td>Sort dimension members for Entity, Account, Scenario, Versions, and user-defined custom dimensions.</td>
<td>&quot;Working with Applications that Use Planning Application Administration&quot; on page 359</td>
</tr>
<tr>
<td>TaskListDefUtil</td>
<td>Move task list definitions between Planning applications by exporting or importing task list definitions to or from an XML file.</td>
<td>&quot;Importing and Exporting Task Lists&quot; on page 268</td>
</tr>
</tbody>
</table>
Suppressing Password Prompts in Planning Utilities

When running 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” on page 53).

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 Planning utilities use text files that must be saved in UTF-8 format: `FormDefUtil`, `SampleApp_Data`, and `TaskListDefUtil`. 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 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, 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 are logged to a log file called `PlanningCLU.log` that is generated under `EPM_ORACLE_INSTANCE/diagnostics/logs/planning`. The debug level can be modified in the `loggingCLU.xml` file located under `EPM_ORACLE_INSTANCE/Planning/planning1`.

Using Application Servers

You can access 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 Essbase Server:
  1. Maximize the Essbase window.
  2. Enter Quit.

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

- To close Planning on the Web, select File, then Exit.
Levels of Access Permissions

Setting access permissions to 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.
  - Dimension Editor: Required for Performance Management Architect and Planning application administration. For Performance Management Architect, allows access to application administration options for Planning. For Planning application
administration, allows access to the Planning application administration options for Planning (in combination with the Planning Application Creator role).

- Planning Application Creator: Required for Performance Management Architect and Planning application administration. For Performance Management Architect, allows users to create Planning applications and Performance Management Architect Generic applications. For Planning application administration, allows access to the Planning Application Administration options for Planning (in combination with the Dimension Editor role). For detailed information on these roles, see the Oracle Enterprise Performance Management System User Security Administration Guide.

- 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 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 Essbase database. After the database is refreshed, appropriate access permissions are applied to the user.

### Types of Access Permissions

Access permissions for the specified user or group to the dimension member, form, or task list include:
- **Read**: Allow view access
- **Write**: Allow view and modify access
- **None**: Prohibit access; the default access is None

You can also set who can launch which Calculation Manager business rules as:

- **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. Access permissions assigned to members take precedence over inherited access permissions. You can include or exclude the member from the access permissions setting.

#### Table 19  Options for Inheriting Access Permissions

<table>
<thead>
<tr>
<th>Inheritance Option</th>
<th>Access Permission Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Member</td>
<td>Only to the currently selected member.</td>
</tr>
<tr>
<td>Children</td>
<td>To all children members in the level below the currently selected member.</td>
</tr>
<tr>
<td>iChildren</td>
<td>To the currently selected member and all children members in the level below it.</td>
</tr>
<tr>
<td>Descendant</td>
<td>To all descendant members below the currently selected member.</td>
</tr>
<tr>
<td>iDescendant</td>
<td>To the currently selected member and all descendant members below it.</td>
</tr>
</tbody>
</table>

#### How access permissions are evaluated

When evaluating access permissions, 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**, then **Dimensions**.
2. For **Dimension**, select the dimension to change.
3. Select **Actions**, and then select **Edit**.
4. In **Dimension Properties**, select **Apply Security** to allow access permissions to be set on its members.

**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” on page 58.

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” on page 58.

➢ To assign access to members or business rules:

1. Select the member or business rule:
   - For members: Select **Administration**, then **Manage**, then **Dimensions**, then select the dimension and member.
   - For business rules: Select **Administration**, then **Business Rule Security**. Select the folder containing the business rules, 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 Shared Services Console to 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.**


### Adding Access

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

1. **Select the member or business rule:**
   - For members: Select *Administration*, then *Manage*, then *Dimensions.* Then select the dimension and member.
   - For Calculation Manager business rules: Select *Administration*, then *Business Rule Security.* Select the folder containing the business rules, 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 Shared Services Console to 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. Click *Add Access.*

6. 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.

7. **Optional for members:** Select a relationship.

   For example, select *Children* to assign access to the children of the selected member.

8. **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.

9. **For the selected users or groups,** select the access type and click *Add.*
10 Click Close.

**Editing Access**

You can access the **Assign Access** page for a member by selecting **Assign Access** on **Dimensions** tab

➤ To modify access permissions for members or Calculation Manager business rules:

1 Select the member or business rule:
   - For members: Select **Administration**, then **Manage**, then **Dimensions**, then select the dimension and member.
   - For Calculation Manager business rules: Select **Administration**, then **Business Rule Security**. Select the folder containing the business rules, 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 Shared Services Console to 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 Click **Edit Access**.

6 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.

7 **Optional for members**: Select a relationship.

   For example, select Children to assign access to children of the selected member.

8 Click **Set**.

9 Click **Close**.

**Deleting Access**

➤ To remove access permissions for members or Calculation Manager business rules:

1 Select the member or business rule:
   - For members: Select **Administration**, then **Manage**, then **Dimensions**, then select the dimension and member.
For Calculation Manager business rules: Select Administration, then Business Rule Security. Select the folder containing the business rules, 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 Shared Services Console to 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 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.

6 Click Remove Access.

7 Click OK.

8 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.

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” on page 83).

Sample Entity Members

<table>
<thead>
<tr>
<th>Parent Entity</th>
<th>Child Entities</th>
</tr>
</thead>
<tbody>
<tr>
<td>United States</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CA (base)</td>
</tr>
<tr>
<td></td>
<td>NY</td>
</tr>
<tr>
<td>West</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CA (shared)</td>
</tr>
<tr>
<td></td>
<td>NV</td>
</tr>
<tr>
<td>Sales Region 1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CA (shared)</td>
</tr>
</tbody>
</table>
### Table 20  Example of Inherited Access to Shared Members

<table>
<thead>
<tr>
<th>Case</th>
<th>Access Permission</th>
<th>Effective Access for Base and Shared Member CA</th>
<th>Explanation</th>
</tr>
</thead>
</table>
| 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

**Subtopics**

- Assigning Access to Forms and Folders
- Adding Access to Forms and Folders
- Changing Access to Forms and Folders

### Assigning Access to Forms and Folders

Administrators can assign access to forms, form folders, and Calculation Manager business rule folders. (For information on assigning access to business rules and members, see “Types of Access Permissions” on page 56 and “Assigning Access to Members and Business Rules” on page 58.)

**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 Access to Forms and Folders” on page 63.

**Adding Access to Forms and Folders**

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

1. Select the form or folder.
   - For forms and folders, see “Selecting and Opening Forms and Folders” on page 154.
   - For business rule folders, select **Administration**, then **Manage**, then **Business Rule Security**.
     - For business rules, select **Administration**, then **Manage**, 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 Shared Services Console to 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. **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.
Changing Access to Forms and Folders

To change which users can use or change forms or folders:

1. Select the form or folder.
2. Click Assign Access.
3. Optional: To migrate a user or group's changed identity or their position in the user directory from Shared Services Console to 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. Select the users or groups for which to change access, and click Edit Access.
6. For Type of Access, select the kind of access users or groups have to the form or folder.
7. Click Set.
8. Click Close.

To remove access from forms or folders:

1. Select the form or folder.
2. For forms and folders, see “Selecting and Opening Forms and Folders” on page 154.
3. For Calculation Manager business rule folders, select Administration, then Manage, then Business Rule Security.
- For Calculation Manager business rules, select **Administration**, then **Manage**, then **Business Rule Security**. Open the business rule folder containing the business rules and select the rules.

2. Click **Assign Access**.

3. Optional: To migrate a user or group’s changed identity or their position in the user directory from Shared Services Console to 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. 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.

6. Click **OK**.

### Importing Access Permissions

The **ImportSecurity** utility loads access permissions for users or groups from a text file into 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, 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” on page 68.

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” on page 53).

- 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” on page 58).

- 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, semicolon (;), pipe (|), colon (:), space ( ). Comma is the default.

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>username or group name</td>
<td>The name of a user or group defined in Shared Services Console. 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: sl_group. For example: admin,member1,READ,MEMBER admin,member1,READ,MEMBER,SL_GROUP</td>
</tr>
<tr>
<td>artifact name</td>
<td>The named artifact for the imported access permissions (for example the member, form, task list, folder, or Calculation Manager business rule). Example: Account1. If an artifact name contains a character that you are using as the delimiter, enclose the name in double quotes. For example, if you are using a space as the delimiter, enclose the name South America in double quotes: “South America”.</td>
</tr>
<tr>
<td>access permissions</td>
<td>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: User1,Member1,READ,@ICHILDREN User1,Member1,READWRITE,@ICHILDREN Access permissions for User1 to Member1 are applied as READWRITE. For Calculation Manager business rules and folders only: specify launch access permissions as either NONE or LAUNCH.</td>
</tr>
<tr>
<td>Essbase access flags</td>
<td>@CHILDREN, @ICHILDREN, @DESCENDANTS, @DESCENDANTS and MEMBER. Security implementation for these functions is identical to Essbase. Note: For task lists, only MEMBER can be used. For folders, only @DESCENDANTS can be used.</td>
</tr>
</tbody>
</table>
| artifact type         | For artifacts other than members, distinguish which artifact you are importing security for with artifact type identifier:  
- 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_CALCFOLDER—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
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” on page 53).

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:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[-f:passwordFile]</td>
<td>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” on page 52.</td>
</tr>
<tr>
<td>appname</td>
<td>Name of the Planning application to which you are importing access permissions.</td>
</tr>
<tr>
<td>username</td>
<td>Planning administrator user name.</td>
</tr>
<tr>
<td>delimiter</td>
<td>Optional: SL_TAB, SL_COMMA, SL_PIPE, SL_SPACE, SL_COLON, SL_SEMI-COLON. If no delimiter is specified, comma is the default.</td>
</tr>
<tr>
<td>RUN_SILENT</td>
<td>Optional: Execute the utility silently (the default) or with progress messages. Specify 0 for messages, or 1 for no messages.</td>
</tr>
<tr>
<td>[SL_CLEARALL]</td>
<td>Optional: Clear existing access permissions when importing new access permissions. Must be in uppercase.</td>
</tr>
</tbody>
</table>

For example:

   `ImportSecurity "app1,admin,SL_TAB,1"

To clear all access permissions, enter:

   `ImportSecurity "app1,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” on page 53.
Improving Performance When Importing Access Permissions

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

1 On 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 Planning access permissions to the SecFile.txt file, enabling you to export and import access permissions across applications (see “Importing Access Permissions” on page 65). 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” on page 53).
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:
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Mandatory?</th>
</tr>
</thead>
<tbody>
<tr>
<td>[-f:passwordFile]</td>
<td><strong>Optional</strong>: 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” on page 52.</td>
<td>No</td>
</tr>
<tr>
<td>/A=appname</td>
<td>The name of the Planning application from which you are exporting access permissions.</td>
<td>Yes</td>
</tr>
<tr>
<td>/U=username</td>
<td>The administrator's ID for logging into the application.</td>
<td>Yes</td>
</tr>
<tr>
<td>/S=searchCriteria</td>
<td>The user or group name.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>You cannot use this option with <code>/S_USER</code> or <code>/S_GROUP</code>.</td>
<td></td>
</tr>
<tr>
<td>/S_USER=user</td>
<td>A specified user name.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>You cannot specify multiple users or use this option with <code>/S_GROUP</code> or <code>/S=searchCriteria</code>.</td>
<td></td>
</tr>
<tr>
<td>/S_GROUP=group</td>
<td>A specified group. Only matching groups, not matching user names, are exported.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>You cannot specify multiple groups or use this option with <code>/S_USER</code> or <code>/S=searchCriteria</code>.</td>
<td></td>
</tr>
<tr>
<td>/S_MEMBER=MemberName</td>
<td>A specified member.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>You can specify only one member-based parameter.</td>
<td></td>
</tr>
<tr>
<td>/S_MEMBER_ID=MemberName</td>
<td>A specified member and its descendants.</td>
<td>No</td>
</tr>
<tr>
<td>/S_MEMBER_D=MemberName</td>
<td>A specified member's descendants.</td>
<td>No</td>
</tr>
<tr>
<td>/S_MEMBER_IC=MemberName</td>
<td>A specified member and its children.</td>
<td>No</td>
</tr>
<tr>
<td>/S_MEMBER_C=MemberName</td>
<td>A specified member's children.</td>
<td>No</td>
</tr>
<tr>
<td>/DELIM=delim</td>
<td><code>SL_TAB, SL_COMMA, SL_PIPE, SL_SPACE, SL_COLON, SL_SEMI-COLON</code>. If no delimiter is specified, comma is the default.</td>
<td>No</td>
</tr>
<tr>
<td>/DEBUG=</td>
<td>Specify <code>true</code> to display the utility's performed steps. <code>false</code> is the default.</td>
<td>No</td>
</tr>
<tr>
<td>/TO_FILE=</td>
<td>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” on page 53). If you specify another path, use double backslashes, for example: <code>C:\Oracle \SecFile.txt</code>.</td>
<td>No</td>
</tr>
<tr>
<td>/HELP=Y</td>
<td>Specify as the only parameter to display the syntax and options for <code>ExportSecurity</code>.</td>
<td>No</td>
</tr>
</tbody>
</table>

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 quotes. For example, if a space is the delimiter, the name South America is enclosed in double quotes: “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:

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user or group</td>
<td>The name of a user or group defined in Shared Services Console.</td>
</tr>
<tr>
<td>memName</td>
<td>A member in the application.</td>
</tr>
<tr>
<td>access permissions</td>
<td>READ, READWRITE, or NONE. If there are duplicate lines for a user name/member name combination, the line with READWRITE access takes precedence. For Calculation Manager business rules and folders only: Access permissions are specified as either NONE or LAUNCH.</td>
</tr>
<tr>
<td>Essbase access flags</td>
<td>@CHILDREN, @ICHILDREN, @DESCENDANTS, @IDESCENDANTS, and MEMBER. Security implementation for these functions is identical to Essbase.</td>
</tr>
<tr>
<td>artifact type</td>
<td>After each line, the utility appends the artifact type:</td>
</tr>
<tr>
<td></td>
<td>- SL_FORM— for forms</td>
</tr>
<tr>
<td></td>
<td>- SL_COMPOSITE— for composite forms</td>
</tr>
<tr>
<td></td>
<td>- SL_TASKLIST— for task lists</td>
</tr>
<tr>
<td></td>
<td>- SL_CALCRULE— for business rules</td>
</tr>
<tr>
<td></td>
<td>- SL_FORMFOLDER— for form folders</td>
</tr>
<tr>
<td></td>
<td>- SL_CALCFOLDER— for folders containing business rules</td>
</tr>
</tbody>
</table>

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
Reporting on Access Permissions

Subtopics

- Selecting Reporting Objects
- Selecting Reporting Options
- Working With Access Permissions Reports
- Setting up Audit Trails
- Examples of Tracked Actions
- Viewing and Clearing Audit Reports

You can view current access permissions and print reports.

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

1. In Shared Services Console, select a Planning application under Application Groups. Select Administration, then Access Control Report.

2. On Select User or Group, select options:
   - Available Users
   - Available Groups
   - Available Users and Groups

3. From the left Available panel, select and move users or groups to report on to the Selected panel:
   - To move selections, click .
   - To remove selections, click .
   - To move all users or groups, click .
   - To remove all users and groups, click .

   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:

See “Reporting on Access Permissions” on page 71.

2 On Select Objects, select the Planning objects on which to report:
   - To move selections to Selected Objects, click ➡.
   - To remove selections, click ◀.
   - To move all objects, click ➡. 
   - To remove all objects, click ◀.

3 Click Next.

Selecting Reporting Options

- To specify options for access reports:
  1 Start the Access Control Report.
      See “Reporting on Access Permissions” on page 71.
  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>
<thead>
<tr>
<th>Report Type</th>
<th>Description</th>
<th>Options</th>
</tr>
</thead>
</table>
| Assigned Access| Summarizes access permissions that administrators assign| Specify whether access is assigned by member selection relation or group membership:  
   - 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 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. |

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>
<thead>
<tr>
<th>Table 22</th>
<th>Actions That Can be Audited</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Audit Options</strong></td>
<td><strong>Tracked Changes</strong></td>
</tr>
</tbody>
</table>
| Dimension Administration | • Dimension hierarchy: adding a member or dimension, moving, deleting, 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 Planning (such as importing form designs with the ImportFormDefinition utility) |
| Alias Table Administration | Changes to alias tables: creating, copying, renaming, deleting, and clearing |
| Data | • Cell values  
• Supporting detail  
• Account annotations  
• Cell-level documents |
| Launch Business Rules | Updates from calc 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 |
| Offline | Forms taken offline or synchronized back to the server |
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, Planning tracks:

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

```sql
DELETE FROM HSP_AUDIT_RECORDS
```

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

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

Managing Security Filters

Access permissions in Planning are stored in the relational database. If you use other products outside of Planning, such as Financial Reporting or third-party tools, to access Planning data directly in 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” on page 83).

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 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 Planning users, groups, and roles in Shared Services Console with a Planning application and with 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” on page 53.

   Table 23  ProvisionUsers Syntax

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Required?</th>
</tr>
</thead>
<tbody>
<tr>
<td>[-f:passwordFile]</td>
<td>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 passwordFile. See “Suppressing Password Prompts in Planning Utilities” on page 52.</td>
<td>No</td>
</tr>
<tr>
<td>/ADMIN:adminName</td>
<td>The administrator's name for logging on to the Planning application.</td>
<td>Yes</td>
</tr>
<tr>
<td>/A:appName</td>
<td>The Planning application to synchronize (must be on the server on which the utility is run).</td>
<td>Yes</td>
</tr>
<tr>
<td>[/U:user1[user2:user3]]</td>
<td>Specifies users to synchronize. For example, to synchronize users Planner1 and Planner2, use /U:Planner1;Planner2. Omitting this argument synchronizes all users.</td>
<td>No</td>
</tr>
<tr>
<td>[/R:n]</td>
<td>Specifies an interval, in minutes, in which synchronization is run. For example, to synchronize every 30 minutes, use /R:30. Omitting this argument performs the synchronization once.</td>
<td>No</td>
</tr>
<tr>
<td>/?</td>
<td>Specified by itself, prints the syntax and options for ProvisionUsers.</td>
<td>No</td>
</tr>
</tbody>
</table>

2. If prompted, enter your password.

   Example 1
   Entering:
   ProvisionUsers /ADMIN:admin /A:App1
   Synchronizes all users in the App1 application.

   Example 2
   Entering:
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 Planning.

To migrate changed user and group identities from 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 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 Business Rules, administrators can migrate launch access permissions on business rules and their projects from Business Rules to Calculation Manager business rules in 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 Shared Services Console.

To migrate access permissions on business rules and their folders:

1. **Before running** **HBRMigrateSecurity.cmd**:
   - Migrate business rules from Oracle Hyperion 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:**

   ```
   ```
For the full path to planning1, see “About EPM Oracle Instance” on page 53.

Table 24  HBRMigrateSecurity Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Purpose</th>
<th>Required?</th>
</tr>
</thead>
<tbody>
<tr>
<td>[-f:passwordFile]</td>
<td>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 passwordFile. See “Suppressing Password Prompts in Planning Utilities” on page 52.</td>
<td>No</td>
</tr>
<tr>
<td>/A:appname</td>
<td>Specify the Planning application to which to migrate launch access permissions for business rules</td>
<td>Yes</td>
</tr>
<tr>
<td>/U:admin</td>
<td>Specify the administrator’s user name</td>
<td>Yes</td>
</tr>
<tr>
<td>/F:output file</td>
<td>Specify the name of the XML output file, including its full path if it is not in the planning1 directory (for the full path to planning1, see “About EPM Oracle Instance” on page 53). This file contains a log of the transaction, and helps with troubleshooting.</td>
<td>Yes</td>
</tr>
<tr>
<td>/?</td>
<td>Print the syntax and options for HBRMigrateSecurity.cmd</td>
<td>No</td>
</tr>
</tbody>
</table>

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” on page 188.

Removing Stale User Records

When you deprovision or delete users or groups in Shared Services, you can conserve disk space by updating the user and group tables in the 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 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

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
For applications using Planning application administration, databases are created and maintained within Planning. For Performance Management Architect applications, applications are created in Performance Management Architect and deployed to Planning. For information on tasks performed in Performance Management Architect, see the Oracle Hyperion Enterprise Performance Management Architect Administrator's Guide.

Unlocking Applications

Occasionally 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” on page 53.

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” on page 52.

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 all Planning users currently 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.

For users logged on to the application, the broadcast message displays in their browser the next time they refresh the page or go to a different page. Only users who are currently logged on to the Planning application see broadcast messages.

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” on page 53.

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” on page 52.

*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:

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

## Viewing Usage Statistics

You can determine which 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 Financial Reporting, Smart View, 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” on page 90).

To view usage statistics:

1 From Planning Web, log on to an application.

2 Select Administration, then Application, then Statistics.

Usage statistics display for each server.

## Creating and Refreshing Application Databases

On the Manage Database page, you can create and refresh application databases, which are used to store data in Planning for each plan type in the application. Databases are structured according to dimensions, hierarchical members, attributes, and other data specified in an application.

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 calc 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” on page 91. 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” on page 36.

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” on page 278.

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” on page 91.
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” on page 75.

  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” on page 278.

  **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” on page 61).

  **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**.

7 **Optional**: If the create or refresh process takes some time, you can click **Run in Background** to run the process in the background without displaying the status.
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” on page 86 and “Editing Exchange Rate Tables” on page 87.

Note: Use Performance Management Architect to specify the base currency, currency code and symbol, triangulation currency, reporting currency, and exchange rate type. See the Oracle Hyperion Enterprise Performance Management Architect Administrator’s Guide.

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.

Note: You can change the density of the HSP_Rates dimension in Performance Management Architect. Planning supports currency conversion by triangulation through the triangulation currency set in Performance Management Architect.

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, then Exchange Rates.
2. Click Create.
3. Specify information for the Exchange Rate table:
Name

Description

Click Save, then define settings for the table as specified in “Editing Exchange Rate Tables” on page 87.

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:

1. Select Administration, then Manage, then Exchange Rates, select the table to edit, then click Edit.

2. In the Rate Table tab, select options:
   - Display Options
   - Average
   - Ending
   - BegBalance
   - Historical
   - Method

3. Click Next.

4. In the Exchange Rate tab, set options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>To Currency</td>
<td>The currency for which to enter conversion rates (the default currency or a triangulation currency).</td>
</tr>
<tr>
<td>Show Years</td>
<td>The time periods displayed (by default, the current application year).</td>
</tr>
<tr>
<td>Rate Table Name</td>
<td>The name of the exchange rate table (display only).</td>
</tr>
<tr>
<td>Method</td>
<td>Multiply or Divide, the mathematical operator that determines how values are calculated between the source and destination currencies.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Historical</td>
<td>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.</td>
</tr>
<tr>
<td>BegBalance</td>
<td>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.</td>
</tr>
<tr>
<td>Avg</td>
<td>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.</td>
</tr>
<tr>
<td>End</td>
<td>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.</td>
</tr>
</tbody>
</table>

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, then click Next.
   - Name
   - Description
In the Details tab, select information for the Currency Conversion Script details:

- Currency
- Scenario
- Version Type: Bottom-up or Target
- Version

Click to select members for fields.

**Working with Currency Conversion Calc Scripts**

If multiple currencies are enabled for the Planning application when you create a currency conversion, a currency conversion calc script is created, based on selected scenarios, versions, and currencies. A second calc 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 calc script is named HspCrtB.csc for bottom-up versions and HspCrtT.csc for target versions. Running the copy calc script enables the currency conversion calc 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 calc 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 calc script and the currency conversion calc script. After launching the HspCrtB.csc or HspCrtT.csc copy calc 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 calc script for calc scripts, you must regenerate currency conversion calc scripts.

**Optimizing Performance**

Subtopics

- About Reordering Dimensions
- Allocating Memory for Supporting Detail Cache
- Configuring the Data and Index Cache Sizes
- Other Performance Optimization Tips

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” on page 312, “About Reordering Dimensions” on page 90, and “Setting Dimension Density and Order” on page 325).

• Design business rules to execute within two minutes. Schedule longer-running business rules to execute in the background (see “Setting Background Processing” on page 44).

• Set the **Suppress Missing Blocks** option (see “Setting Form Grid Properties” on page 162).

• Set the 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” on page 39).

• Optimize the JDBC connection pool settings (see “Configuring JDBC Connection Pooling” on page 41).

• Increase the timeout value for SQL queries (see “Optimizing SQL Relational Databases” on page 38).

• If you use the WebLogic Server, check its performance settings (see “Optimizing WebLogic Server Parameters” on page 401).

• Tune Java Virtual Machine (JVM) parameters (see “500 Error Message” on page 398).

• Optimize Windows network parameters (see “Optimizing Windows Network Parameters” on page 402).

### About Reordering Dimensions

The order of dimensions is critical for the structure and performance of a 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.

You can also use Performance Management Architect to change the order for calculating dimensions. See the *Oracle Hyperion Enterprise Performance Management Architect Administrator’s Guide*.

### 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” on page 83. To change its value, see “Setting Application and System Properties” on page 39.

**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 Administration Services or manually using MaxL statements. For example, using MaxL:

```
alter databaseName planType set data_cache_size 20000MB
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 Essbase database.
- Split large forms into multiple smaller forms having fewer rows and columns. Organize related forms into folders.
- Using account annotations impacts performance, so use them sparingly.
- The first time 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 Essbase
- Backing up the relational database for Planning and Financial Reporting
- Backing up required components of Planning

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

Subtopics

- About Loading Dynamic Members
- Loading Tools

About Loading Dynamic Members

Child members that you load using either of the Outline Load Utilities or Lifecycle Management under parent members enabled for dynamic children, are added as dynamic child members if there are dynamic member placeholders in Essbase. Once 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” on page 326.

Loading Tools

You can use the tools identified in the following tables to load metadata and data.
Table 26  Metadata Load Tools

<table>
<thead>
<tr>
<th>Metadata Load Tool</th>
<th>For Planning Application Administration</th>
<th>For Performance Management Architect</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outline Load utility</td>
<td>X</td>
<td></td>
<td>See “Working with the Outline Load Utility” on page 95.</td>
</tr>
<tr>
<td>Planning Outline Load</td>
<td>X</td>
<td></td>
<td>See “Working with Planning Outline Load” on page 140.</td>
</tr>
<tr>
<td>Oracle Data Integrator (ODI) Adapter for Planning</td>
<td>X</td>
<td></td>
<td>See the Oracle Hyperion Data Integration Management for Planning User's Guide.</td>
</tr>
<tr>
<td>Data Integration Management (DIM) Adapter for Planning</td>
<td>X</td>
<td></td>
<td>See the Oracle Data Integrator Adapter for Planning Online Help.</td>
</tr>
<tr>
<td>Performance Management Architect flat files</td>
<td>X</td>
<td></td>
<td>See the Oracle Hyperion Enterprise Performance Management Architect Administrator's Guide.</td>
</tr>
<tr>
<td>Performance Management Architect interface tables</td>
<td>X</td>
<td></td>
<td>See the Oracle Hyperion Enterprise Performance Management Architect Administrator's Guide.</td>
</tr>
</tbody>
</table>

Table 27  Data Load Tools

<table>
<thead>
<tr>
<th>Data Load Tool</th>
<th>For Planning Application Administration</th>
<th>For Performance Management Architect</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outline Load utility</td>
<td>X</td>
<td></td>
<td>Loads numeric, date, and text data values. See “Working with the Outline Load Utility” on page 95.</td>
</tr>
<tr>
<td>Planning Outline Load</td>
<td>X</td>
<td></td>
<td>See “Working with Planning Outline Load” on page 140.</td>
</tr>
</tbody>
</table>
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 Outline Load” on page 140. Planning Outline Load 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” on page 95
- “Generating Load Files” on page 97
- “Importing Data and Metadata Using a Flat File” on page 101
- “Importing Data and Metadata from a Relational Data Source” on page 104
- “Exporting Data and Metadata from a Planning Application to a Flat File” on page 111
- “Command Line Parameters for the Outline Load Utility” on page 116
- “Dimension Properties” on page 125

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 run time and the results of the parsing are displayed in the log.

For example, the following command line:

```
```

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`

**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

Subtopics

- Generating a Data Load File
- Generating a Metadata Load File
- Comments in Load Files
- Load File Considerations

Generating a Data Load File

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

**Note:** You can also load to driver members that are specified on the Planning Data Load Administration page.

Generate a comma-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.

```
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 .CSV 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:

```
<table>
<thead>
<tr>
<th>Currency</th>
<th>Version</th>
<th>Scenario</th>
<th>Year</th>
<th>Entity</th>
<th>Period</th>
<th>Account</th>
<th>HSP_Rates</th>
</tr>
</thead>
<tbody>
<tr>
<td>USD</td>
<td>BUVersion_1</td>
<td>Current</td>
<td>FY07</td>
<td>e1</td>
<td>Jan</td>
<td>aUnspec</td>
<td>HSP_InputValue</td>
</tr>
<tr>
<td>USD</td>
<td>BUVersion_1</td>
<td>Current</td>
<td>FY07</td>
<td>e1</td>
<td>Jan</td>
<td>aText</td>
<td>HSP_InputValue</td>
</tr>
<tr>
<td>20081222</td>
<td>USD</td>
<td>BUVersion_1</td>
<td>Current</td>
<td>FY07</td>
<td>e1</td>
<td>Date</td>
<td>HSP_InputValue</td>
</tr>
<tr>
<td>1</td>
<td>USD</td>
<td>BUVersion_1</td>
<td>Current</td>
<td>FY07</td>
<td>e1</td>
<td>Jan</td>
<td>aSmart</td>
</tr>
</tbody>
</table>
```

**Example 3**: Specify drivers directly in the .CSV 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
```


To add comments to a .CSV load file, see “Comments in Load Files” on page 99.

### 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 .CSV load file contain metadata records, listed in the order designated by the header record. Each metadata record contains a comma-separated list of property values that correspond to the header record entries. For detailed information on the properties available for each Planning member, see “Dimension Properties” on page 125.

**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
el, 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 .CSV load file, see “Comments in Load Files” on page 99.

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, Planning internal, comment block header.
Load File Considerations

Consider these points when working with load files:

- If saving or modifying metadata or data .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 quotes 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 quotes 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” on page 125.

- 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 Planning application (for example, to change a property), if the member already exists in the application 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>
<thead>
<tr>
<th>Value</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;quote&quot;&quot;quote&quot;</td>
<td>quote*quote</td>
</tr>
<tr>
<td>&quot;&quot;quotedstring&quot;&quot;</td>
<td>&quot;quotedstring&quot;</td>
</tr>
<tr>
<td>&quot;&quot;quoted, *, string,&quot;&quot;</td>
<td>&quot;,quoted,&quot;string,&quot;</td>
</tr>
<tr>
<td>&quot;&quot;quoted, <strong>, string,</strong>&quot;</td>
<td>&quot;,quoted,&quot;string,&quot;</td>
</tr>
</tbody>
</table>

**Table 28 Examples of Data Values Containing Commas and Quotation Marks**

**Importing Data and Metadata Using a Flat File**

**Subtopics**

- **Loading Data**
- **Loading Metadata**

**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 Planning Data Load Administration page, or you can specify driver members in the .CSV load file and run the utility with the /TR option.

If you load data with the /TR option, the .CSV 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 .CSV file. You can include multiple rows, but you can specify only one data value per row. See “Generating Load Files” on page 97.

---

**Caution!** Following these steps can affect data in the database. The Planning DIRECT_DATA_LOAD system property enables data to be loaded directly to 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 .CSV 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 .CSV 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 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” on page 97.

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” on page 53.
   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 .CSV load file parses successfully. For example, you could use this command line to check the load file for a Planning application called test:
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 /N /I:c:\outline1data3.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:\outline1data3.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” on page 125.

### Loading Metadata

Metadata for applications using 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 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” on page 98.
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” on page 53.
   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:\outline1_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” on page 125.

## Importing Data and Metadata from a Relational Data Source

### Subtopics

- Importing Metadata
- Importing Data

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 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” on page 116.

**Caution!** Before performing an import operation, ensure you back up the Planning relational store and the 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” on page 116.

**Example: The myprop_relational.properties File**

```
/S:localhost

/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.DESCRIPTION = 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.DESCRIPTION = 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
/RIU:PS2ORAU
/RIP:password
```

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: 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: 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)

/RQ: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:
  
<table>
<thead>
<tr>
<th>Member</th>
<th>Parent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ex</td>
<td>Entity</td>
</tr>
</tbody>
</table>

- 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: Using the /C2A Parameter for Importing Aliases into an Alias Table
/A:TARGET
/U:admin
/IR
/D:Entity

/RQ:ENT_ALIAS_ASSIGNMENT_QUERY
/C2A:(AliasTableName, Alias: LongAliasTableName012345678901234567890123456789)

ENT_ALIAS_ASSIGNMENT_QUERY=SELECT O.OBJECT_NAME as Entity, (select object_name from
Note: In the previous example, both the target Planning application and the source contain a long alias table name.

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

Subtopics

- Clearing an Essbase Block Using the /ICB Parameter
- Setting Outline Load Utility Drivers Using the /SDM Parameter

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 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” on page 109.

To import data from a relational database:

1. Backup the 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:

3. Create a Planning application form.

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


   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
   ```
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:

![Planning application form](image)

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 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:

![Essbase log file](image)

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

![Planning form](image)

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,aATDT,aENTITY,aPOIN
```

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

The following example is a relational table:

<table>
<thead>
<tr>
<th>Account</th>
<th>Data Load Cube Name</th>
<th>Point-of-View</th>
<th>Jan</th>
<th>Feb</th>
<th>Mar</th>
</tr>
</thead>
<tbody>
<tr>
<td>aUnspec</td>
<td>Plan1</td>
<td>FY12, Ron,Curr</td>
<td>10</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>aPer</td>
<td>Plan1</td>
<td>FY12, Ron,Curr</td>
<td>0.65</td>
<td>0.65</td>
<td></td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>

The resulting Planning application form:
Exporting Data and Metadata from a Planning Application to a Flat File

Subtopics

- Exporting Data with the Outline Load Utility
- Exporting Metadata with the Outline Load Utility

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 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:

```
/S:localhost
/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” on page 116.

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.

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:
Exporting Metadata with the Outline Load Utility

You can use the Outline Load utility to export metadata for a dimension to a .CSV 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” on page 116.

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:pln1dv /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:

```
```

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
```

Loading Data and Metadata
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.

**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 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” on page 116.

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:

<table>
<thead>
<tr>
<th></th>
<th>E1</th>
<th>e1's description</th>
<th>Entity</th>
</tr>
</thead>
<tbody>
<tr>
<td>E11</td>
<td>e2's description</td>
<td>e1</td>
<td></td>
</tr>
</tbody>
</table>

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:

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

**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: 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 Entity E1 Alias From 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.

```
[/CP:commandPropertieFileName] [/M] [/I:inputFileName]/
IR[:RDBConnectionPropertiesFileName]/IRA|/E:outputFileName/
ED:outputFileNameStem] [/ED:dataProviderSpecification] [/C]
D[U]:loadDimensionName|/DA:attributeDimensionName:baseDimensionName|TR]
RIQ:inputQueryOrKey] [/RIC:catalog] /RID:driver /RIR:url /RIU:userName [/]
RIP:password] [/X:exceptionFileName] [L:logFileName] [/?]
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[-f:passwordFile]</td>
<td><strong>Optional</strong>: 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>.</td>
</tr>
<tr>
<td>/S:server</td>
<td>If specified, the server into which you are currently logged in. If not specified, localhost is used.</td>
</tr>
<tr>
<td>/A:application</td>
<td>Name of the Planning application to which you are importing.</td>
</tr>
<tr>
<td>/U:userName</td>
<td>User name with which to log on to the application.</td>
</tr>
<tr>
<td>/CP:commandPropertieFileName</td>
<td>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” on page 95.</td>
</tr>
<tr>
<td>/M</td>
<td>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).</td>
</tr>
<tr>
<td>/I:inputFileName</td>
<td>Specifies the data load input file which contains a header record and data records in <code>.csv</code> 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 Essbase data.</td>
</tr>
<tr>
<td>/IR[:RDBConnectionPropertiesFileName]</td>
<td>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.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>/IRA</td>
<td>Identical to the /IR switch except that the required RDB JDBC connection properties (/RIQ, /RIC, /RID, /RIR, /RIU, /RIP switch values) are obtained from the currently connected application's RDB data source. A data load dimension (/D option) must be specified as well. The /ICB switch can be specified to clear Essbase data.</td>
</tr>
</tbody>
</table>
| /ICB:blockSpecification | Clears an Essbase block before performing an import operation (/I, /IR, /IRA). (A string of the form \\
<p>| /ALS            | Create alias tables on import if they do not exist (default). Use /-ALS to error out if the referenced alias tables do not exist.                                                                               |
| /B:outputFileName | Exports the dimension specified with the /D switch to the specified output file. (When exporting planning unit hierarchies, the file is in the format defined for importing planning unit hierarchies.)                     |
| /ED:outputFileNameStem | Exports data to Planning Driver Member formatted data files. The /EDD switch must also be set. Files are generated in the form: &lt;fileNameStem&gt;.1-n.csv through &lt;fileNameStem&gt;.n-n.csv where n is the number of files generated. |
| /EDH            | Exports a dimension header in Planning internal HEADERBLOCK format in the output .CSV file. This is used on import to dynamically create base and attribute dimensions prior to import.                           |
| /ER[:RDBConnectionPropertiesFileName] | Specifies that export records will be written to a relational database table. Setting the /EDD 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 (/REQ, /REC, /RED, /RER, /REU, and /REP) can be found within the properties file. A data load dimension must also be specified using the /D switch. |
| /ERA            | Identical to the /ER switch except that the required RDB JDBC connection properties (/REQ, /REC, /RED, /RER, /REU, /REP switch values) are obtained from the currently connected application's RDB data source. Setting the /EDD switch specifies that data will be exported. A data load dimension must also be specified using the /D switch. |
| /EDD:dataExportSpecification | Specifies the format of the exported Planning Driver Member formatted data files. A string of the form &lt;loadDimensionMembers,...&gt;, &lt;driverMembers,...&gt;, &lt;point-of-view members,...&gt;, &lt;dataLoadCubeName&gt;&quot;. |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/SDM:driverMemberSpecification</td>
<td>Sets driver members on the base load dimension for the application (/A) on import operations only (/I, /IR, /IRA). (A string of the form &quot;&lt;baseLoadDimension&gt;, &quot;&lt;driverMembers,...&gt;, &lt;PlanTypeName&gt;&quot;)</td>
</tr>
<tr>
<td>/D:loadDimensionName</td>
<td>Dimension to be loaded, whose member fields correspond to the header record in the load file. You must also specify a load file (/I) or the planning unit hierarchy to be exported with the /E switch. See the following rows to load user-defined dimensions and attributes using /DU, /DA[T], /DAN, /DAB, and /DAD.</td>
</tr>
<tr>
<td>/DU:userDefinedLoadDimensionName</td>
<td>User-defined dimension to be loaded; a dimension with this name will be created if it does not exist.</td>
</tr>
<tr>
<td>/DA[T]:attributeLoadDimensionName:baseDimensionName</td>
<td>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.</td>
</tr>
<tr>
<td>/DAN:attributeLoadDimensionName:baseDimensionName</td>
<td>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.</td>
</tr>
<tr>
<td>/DAB:attributeLoadDimensionName:baseDimensionName</td>
<td>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.</td>
</tr>
<tr>
<td>/DAD:attributeLoadDimensionName:baseDimensionName</td>
<td>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.</td>
</tr>
<tr>
<td>/DX:HSP_Rates</td>
<td>Load the HSP_Rates dimension and create exchange rate tables if they do not exist.</td>
</tr>
<tr>
<td>/DS:HSP_SMARTLISTS</td>
<td>Load the Smart Lists dimension and Smart List dimension entries.</td>
</tr>
<tr>
<td>/TR</td>
<td>Load data when driver members are specified in the .CSV 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 .CSV file.</td>
</tr>
<tr>
<td>/T</td>
<td>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.</td>
</tr>
<tr>
<td>/N</td>
<td>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).</td>
</tr>
<tr>
<td>/O</td>
<td>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.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>/H</td>
<td>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.</td>
</tr>
<tr>
<td>/R</td>
<td>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. <strong>Note:</strong> Use caution with /R; this option removes attribute bindings and approvals states.</td>
</tr>
<tr>
<td>/DPU</td>
<td>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.</td>
</tr>
<tr>
<td>/C</td>
<td>Perform a cube refresh after the metadata load. Use /-C if you do not want to perform a cube refresh (default). See also /F.</td>
</tr>
<tr>
<td>/F</td>
<td>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.</td>
</tr>
<tr>
<td>/K</td>
<td>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).</td>
</tr>
<tr>
<td>/8</td>
<td>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.</td>
</tr>
</tbody>
</table>
| /DF:datePattern | Override the default date pattern on date data conversions to the specified pattern. The pattern must be one of the following:  
  - MM-DD-YYYY  
  - DD-MM-YYYY  
  - YYYY-MM-DD  
  Use /-DF to use the default date pattern setting (default). |
<p>| /RIQ:inputQueryOrKey | An SQL query or a key in the command arguments properties file (/CP switch) in which the value is a SQL query that is executed to produce input for an import operation. The /IR switch must be set for this to be used, and if it is used, the /RIC, /RID, /RIR, /RIU, and /RIP switches must also be set. |
| /RIC:catalog | An RDB JDBC catalog name for the input RDB connection. The /IR switch must be set for this to be used, and if it is used, the /RIQ, /RID, /RIR, /RIU, and /RIP switches must also be set. If the /IRA switch is used, this switch does not need to be specified. |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/RID:driver</td>
<td>An RDB JDBC driver name for the input RDB connection. The /IR switch must be set for this to be used, and if it is used, the /RIQ, /RIC, /RIR, /RIU, and /RIP switches must also be set. If the /IRA switch is used, this switch does not need to be specified.</td>
</tr>
<tr>
<td>/RIR:url</td>
<td>An RDB JDBC URL for the input RDB connection. The /IR switch must be set for this to be used, and if it is used, the /RIQ, /RIC, /RID, /RIU, and /RIP switches must also be set. If the /IRA switch is used, this switch does not need to be specified.</td>
</tr>
<tr>
<td>/RIU:userName</td>
<td>An RDB JDBC username for the input RDB connection. The /IR switch must be set for this to be used, and if it is used, the /RIQ, /RIC, /RID, /RIU, and /RIP switches must also be set. If the /IRA switch is used, this switch does not need to be specified.</td>
</tr>
<tr>
<td>/RIP:password</td>
<td>An RDB JDBC password for the input RDB connection. The /IR switch must be set for this to be used, and if it is used, the /RIQ, /RIC, /RID, /RIU, and /RIP switches must also be set. If the /IRA switch is used, the /RIP switch need not be specified. Enter the password in its unencrypted form when specifying it for the first time in the .properties file. When the Outline Load utility is run, the properties file will be rewritten with an encrypted value for the /RIP password. If this value is not specified in the properties file, a command line prompt will be issued to obtain the password.</td>
</tr>
</tbody>
</table>
| /REQ:exportQueryOrKey | A SQL query or a key in the command arguments properties file (/CP switch) in which the value is a SQL query that specifies the exported values of the form. `INSERT INTO<tableName> (column1, column2, ...)
VALUES (property1, property2, ...)` where properties are Planning member properties as found in the flat file column header records. The /ER or /ERA switch must be set for this to be used, and if it is used, the /REC, /RED, /RER, /REU, and /REP switches must also be set. |
<p>| /REC:catalog   | An RDB JDBC catalog name for the export RDB connection. The /ER switch must be set for this to be used, and if it is used, the /REQ, /RED, /RER, /REU, and /REP switches must also be set. |
| /RED:driver    | An RDB JDBC driver name for the export RDB connection. The /ER switch must be set for this to be used, and if it is used, the /REQ, /REC, /RER, /REU, and /REP switches must also be set. |
| /RER:url       | An RDB JDBC URL for the export RDB connection. The /ER switch must be set for this to be used, and if it is used, the /REQ, /REC, /RED, /REU, and /REP switches must also be set. |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/REU:userName</td>
<td>An RDB JDBC username for the export RDB connection. The /ER switch must be set for this to be used, and if it is used, the /REQ, /REC, /RED, /RER, and /REP switches must also be set.</td>
</tr>
<tr>
<td>REP:password</td>
<td>An RDB JDBC password for the export RDB connection. The /ER switch must be set for this to be used, and if it is used, the /REQ, /REC, /RED, /RER, and /REU 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.</td>
</tr>
<tr>
<td>/C2A: (column1, alias1), (column2, alias2), ...</td>
<td>Creates aliases for columns for the /RIQ and /REQ commands. Use this parameter to get around the lengthy character column and column alias restrictions on RDBMSs. <strong>Note:</strong> An alias specified on the column overrides assignments made by this command.</td>
</tr>
<tr>
<td>/X: exceptionFileName</td>
<td>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 stderr.)</td>
</tr>
<tr>
<td>/L: logFileName</td>
<td>Specify the file that will contain status and informational messages. (If no file name is specified, the information is written to a file called stdout.)</td>
</tr>
<tr>
<td>/?</td>
<td>Display usage text.</td>
</tr>
</tbody>
</table>

**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.)

```
```

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 .CSV 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.

```
```

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
a11,a1,1

Example: Load a .CSV 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.

Dimension,UDA
Account,New2

Example: Load a .CSV 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 .CSV 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:


Example: /O parameter and .CSV load file order

In the following .CSV 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.

**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.

**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.

**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**


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.

```
```

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:pln1dv /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.

```
```

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, a11, 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.

**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 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

- **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.

- **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” on page 125.

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 29  Account Dimension Properties

<table>
<thead>
<tr>
<th>Column Header in .CSV Load File</th>
<th>Planning Property</th>
<th>Value</th>
<th>Default</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Account</td>
<td>Member Name</td>
<td>Text, subject to member naming restrictions</td>
<td>None</td>
<td>Yes</td>
</tr>
<tr>
<td>Parent</td>
<td>Member Name</td>
<td>Text, subject to member naming restrictions.</td>
<td>None; the name of an existing member, or, if empty, the member is placed as a child under the dimension root</td>
<td>No (Need to specify for most Period members)</td>
</tr>
<tr>
<td>Column Header in .CSV Load File</td>
<td>Planning Property</td>
<td>Value</td>
<td>Default</td>
<td>Required</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-------------------</td>
<td>----------------------------------------------------------------------</td>
<td>---------</td>
<td>----------</td>
</tr>
<tr>
<td>Alias:Alias_Table_Name</td>
<td>Alias</td>
<td>Text, subject to Planning member naming restrictions and aliases already defined in the alias table specified in the column header; &lt;none&gt; removes any alias binding for the member from the specified table.</td>
<td>None</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td></td>
<td>One column header for each alias table defined on the dimension is displayed. Default is a pre-defined alias table defined for every dimension.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>N/A</td>
<td>N/A</td>
<td>No</td>
</tr>
<tr>
<td>Valid For Consolidations</td>
<td>Not Used</td>
<td>N/A</td>
<td>N/A</td>
<td>No</td>
</tr>
<tr>
<td>Data Storage</td>
<td>Data Storage</td>
<td>Text: Store, Dynamic Calc and Store, Dynamic Calc, Never Share, Shared, Label Only</td>
<td>Inherited from the parent. If the parent is the root member, the default is Never Share</td>
<td>No</td>
</tr>
<tr>
<td>Two Pass Calculation</td>
<td>Two Pass Calculation</td>
<td>True, False, or an integer: non-zero is true; zero is false</td>
<td>Inherited from the parent</td>
<td>No</td>
</tr>
<tr>
<td>Description</td>
<td>Description</td>
<td>Text, subject to Planning maximum character restrictions; &lt;none&gt; removes a description</td>
<td>None</td>
<td>No</td>
</tr>
<tr>
<td>Formula</td>
<td>Formula</td>
<td>Text, subject to Essbase restrictions for valid formula syntax; &lt;none&gt; removes a formula</td>
<td>None</td>
<td>No</td>
</tr>
<tr>
<td>UDA</td>
<td>UDA</td>
<td>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; &lt;none&gt; removes all existing UDA bindings.</td>
<td>None</td>
<td>No</td>
</tr>
<tr>
<td>Smart List</td>
<td>Smart Lists</td>
<td>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; &lt;none&gt; removes any existing Smart List binding</td>
<td>None</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td></td>
<td>None</td>
<td>N/A</td>
<td>No</td>
</tr>
<tr>
<td>Data Type</td>
<td>Data Type</td>
<td>Text: Unspecified, Currency, Non-Currency, Percentage, Smart List, Date, Text</td>
<td>Inherited from the Parent; Currency if the member is added under the root dimension</td>
<td>No</td>
</tr>
<tr>
<td>Operation</td>
<td></td>
<td>Text: update; delete level 0; delete idescendants; delete descendants</td>
<td>Update</td>
<td>No</td>
</tr>
<tr>
<td>Account Type</td>
<td>Account Type</td>
<td>Text: Expense, Revenue, Asset, Liability, Equity, Saved Assumption</td>
<td>Inherited from the parent; Revenue if the member is added under the root dimension.</td>
<td>No</td>
</tr>
<tr>
<td>Column Header in .CSV Load File</td>
<td>Planning Property</td>
<td>Value</td>
<td>Default</td>
<td>Required</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>-------------------</td>
<td>-------</td>
<td>---------</td>
<td>----------</td>
</tr>
<tr>
<td>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.)</td>
<td>Weekly Distribution (If Weekly Distribution is set to Even for the application, no distribution option displays.)</td>
<td>True, False, or an integer; nonzero is true; zero is false</td>
<td>None</td>
<td>No</td>
</tr>
<tr>
<td>Skip Value</td>
<td></td>
<td>Text: None, Missing, Zeros, Missing and Zeros; must be None if Account Type is Expense or Revenue</td>
<td>Inherited from the parent</td>
<td>No</td>
</tr>
<tr>
<td>Exchange Rate Type</td>
<td>Exchange Rate Type</td>
<td>Text: None, Average, Ending, Historical (None should not be specified if Data Type is set to Currency; otherwise, it should be specified)</td>
<td>Inherited from the parent; Average if member is added under root dimension.</td>
<td>No</td>
</tr>
<tr>
<td>Variance Reporting</td>
<td>Variance Reporting</td>
<td>Text: Non-Expense, Expense. (Must be Expense if Account Type is Expense; must be Non-Expense for all other Account types.</td>
<td>Inherited from the parent; Non-Expense if the member is added under the root dimension.</td>
<td>No</td>
</tr>
<tr>
<td>Source Plan Type</td>
<td>Source Plan Type</td>
<td>Text: Plan Type names defined in the application (for example, Plan1 or Plan2)</td>
<td>Plan1 or the name of the first plan type defined in the application</td>
<td>No</td>
</tr>
<tr>
<td>Plan Type (Plan1)</td>
<td>Plan Type</td>
<td>True, False, or an integer; non-zero is true; zero is false</td>
<td>Inherited from the parent</td>
<td>No</td>
</tr>
<tr>
<td>Aggregation (Plan1)</td>
<td>Aggregation</td>
<td>Text: +, -, *, /, %, -, Never</td>
<td>Inherited from the parent. If the parent is a root member, the default is +; for Year, the default is ~ (ignore)</td>
<td>No</td>
</tr>
<tr>
<td>Plan Type (Plan2)</td>
<td>Plan Type</td>
<td>True, False, or an integer; non-zero is true; zero is false</td>
<td>Inherited from the parent</td>
<td>No</td>
</tr>
<tr>
<td>Aggregation (Plan2)</td>
<td>Aggregation</td>
<td>Text: +, -, */%-, Never</td>
<td>Inherited from the parent; if the parent is the root member, the default is +; for Year, the default is ~ (ignore)</td>
<td>No</td>
</tr>
<tr>
<td>Column Header in .CSV Load File</td>
<td>Planning Property</td>
<td>Value</td>
<td>Default</td>
<td>Required</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>------------------</td>
<td>-------</td>
<td>---------</td>
<td>----------</td>
</tr>
<tr>
<td>Plan Type (Plan3)</td>
<td>Plan Type</td>
<td>True, False, or an integer: non-zero is true; zero is false</td>
<td>Inherited from the parent</td>
<td>No</td>
</tr>
<tr>
<td>Aggregation (Plan3)</td>
<td>Aggregation</td>
<td>Text: +, -, */%, ~, Never</td>
<td>Inherited from the parent; if the parent is a root member, the default is +; for Year, the default is ~ (ignore)</td>
<td>No</td>
</tr>
<tr>
<td>Attribute Dimension Name (this property is available for sparse Account, Entity, or user-defined dimensions)</td>
<td>Attributes</td>
<td>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; &lt;none&gt; removes all existing attribute bindings for the member; one column header is displayed for each Attribute dimension defined on the dimension.</td>
<td>None</td>
<td>No</td>
</tr>
</tbody>
</table>

**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 30  Entity Dimension Properties**

<table>
<thead>
<tr>
<th>Column Header in .CSV Load File</th>
<th>Planning Property</th>
<th>Value</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entity</td>
<td>Member Name</td>
<td>Text, subject to Planning member naming restrictions.</td>
<td>None</td>
</tr>
<tr>
<td>Two Pass Calculation</td>
<td>Two Pass Calculation</td>
<td>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)</td>
<td>Inherited from the parent</td>
</tr>
<tr>
<td>Column Header in .CSV Load File</td>
<td>Planning Property</td>
<td>Value</td>
<td>Default</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>-------------------</td>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>Base Currency</td>
<td>Base Currency</td>
<td>Text, restricted to currency names already defined in the application</td>
<td>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.</td>
</tr>
</tbody>
</table>

**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” on page 127.

**Table 31**  Period Dimension Properties

<table>
<thead>
<tr>
<th>Column Header in .CSV Load File</th>
<th>Planning Property</th>
<th>Value</th>
<th>Default</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Period</td>
<td>Member Name</td>
<td>Text, subject to Planning restrictions on Period member names.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td>Not exposed in Planning</td>
<td>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.</td>
<td>None</td>
<td>Yes</td>
</tr>
<tr>
<td>Column Header in .CSV Load File</td>
<td>Planning Property</td>
<td>Value</td>
<td>Default</td>
<td>Required</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>-------------------</td>
<td>-------</td>
<td>---------</td>
<td>----------</td>
</tr>
<tr>
<td>Start Period</td>
<td>Text (only applicable to summary time periods)</td>
<td>Text, restricted to time periods already defined for the application.</td>
<td>None</td>
<td>Yes</td>
</tr>
<tr>
<td>End Period</td>
<td>Text (only applicable to summary time periods)</td>
<td>Text, restricted to time periods already defined for the application.</td>
<td>None</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**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” on page 127.

**Table 32  User-Defined Dimension Properties**

<table>
<thead>
<tr>
<th>Column Header in .CSV Load File</th>
<th>Planning Property</th>
<th>Value</th>
<th>Default</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>User-defined dimension name</strong></td>
<td>Member Name</td>
<td>Text, subject to Planning member naming restrictions. Name of the user-defined member being added to the dimension.</td>
<td>None</td>
<td>Yes</td>
</tr>
<tr>
<td>Parent</td>
<td>Member Name or root Dimension Name</td>
<td>Text, subject to Planning member naming restrictions.</td>
<td>Root dimension member</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**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 33  Year Dimension Properties

<table>
<thead>
<tr>
<th>Column Header in .CSV Load File</th>
<th>Planning Property</th>
<th>Value</th>
<th>Default</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Year</td>
<td>Year member</td>
<td>FYnn</td>
<td>None</td>
<td>Yes</td>
</tr>
</tbody>
</table>

### 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 End Period.
- 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 34  Scenario Dimension Properties

<table>
<thead>
<tr>
<th>Column Header in .CSV Load File</th>
<th>Planning Property</th>
<th>Value</th>
<th>Default</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scenario</td>
<td>Scenario Name</td>
<td>Text</td>
<td>None</td>
<td>Yes</td>
</tr>
<tr>
<td>Start Year</td>
<td></td>
<td>First FY year defined in the application (as determined by position)</td>
<td>First year in the application</td>
<td>No</td>
</tr>
<tr>
<td>Start Period</td>
<td></td>
<td>First base time period (as determined by position)</td>
<td>First base period in the application</td>
<td>No</td>
</tr>
<tr>
<td>End Year</td>
<td></td>
<td>Last FY year defined in the application (as determined by position)</td>
<td>Last year in the application</td>
<td>No</td>
</tr>
<tr>
<td>End Period</td>
<td></td>
<td>Last base time period (as determined by position)</td>
<td>Last base period in the application</td>
<td>No</td>
</tr>
<tr>
<td>Exchange Table</td>
<td></td>
<td>The name of an exchange rate table defined in the application.</td>
<td>None</td>
<td>No</td>
</tr>
<tr>
<td>Include BegBal</td>
<td></td>
<td>True, False, or an integer: non-zero is true; zero is false</td>
<td>False</td>
<td>No</td>
</tr>
<tr>
<td>Approvals Enabled</td>
<td></td>
<td>True, False, or an integer: non-zero is true; zero is false</td>
<td>False</td>
<td>No</td>
</tr>
</tbody>
</table>
**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>
<thead>
<tr>
<th>Column Header in .CSV Load File</th>
<th>Planning Property</th>
<th>Value</th>
<th>Default</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>Version Name</td>
<td>Text</td>
<td>None</td>
<td>Yes</td>
</tr>
<tr>
<td>Version Type</td>
<td>Version Type</td>
<td>Bottom Up or Target</td>
<td>Bottom Up</td>
<td>No</td>
</tr>
<tr>
<td>Approvals Enabled</td>
<td>Enabled for Approvals</td>
<td>True, False, or an integer: non-zero is true; zero is false</td>
<td>False</td>
<td>No</td>
</tr>
</tbody>
</table>

**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 Planning.

<table>
<thead>
<tr>
<th>Column Header in .CSV Load File</th>
<th>Planning Property</th>
<th>Value</th>
<th>Default</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Currency</td>
<td>Currency Name</td>
<td>Text</td>
<td>None</td>
<td>Yes</td>
</tr>
<tr>
<td>Symbol</td>
<td>Symbol</td>
<td>Text, subject to Planning currency symbol naming restrictions</td>
<td>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</td>
<td>No</td>
</tr>
<tr>
<td>Scale</td>
<td>Scale</td>
<td>An integer value from 0 to 9, where 0 corresponds to 1, 1 corresponds to 10, 2 corresponds to 100, and so on</td>
<td>No scaling</td>
<td>No</td>
</tr>
<tr>
<td>Triangulation Currency</td>
<td>Triangulation Currency</td>
<td>A currency defined in the application</td>
<td>None</td>
<td>No</td>
</tr>
<tr>
<td>Reporting Currency</td>
<td>Reporting Currency</td>
<td>True, False, or an integer: non-zero is true; zero is false</td>
<td>False</td>
<td>No</td>
</tr>
</tbody>
</table>
Attribute Dimension Properties

Attribute, Parent, Alias: Default, Operation

Notes:

- For properties common to Attribute and Account members, see “Account Dimension Properties” on page 127.
- 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 37  Attribute Dimension Properties

<table>
<thead>
<tr>
<th>Column Header in .CSV Load File</th>
<th>Planning Property</th>
<th>Value</th>
<th>Default</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
<td>Attribute Value Name</td>
<td>Text, subject to Planning member naming restrictions</td>
<td>None</td>
<td>Yes</td>
</tr>
</tbody>
</table>

UDA Dimension Properties

Dimension, UDA, Operation

For properties common to UDA and Account members, see “Account Dimension Properties” on page 127.
### Table 38  UDA Dimension Properties

<table>
<thead>
<tr>
<th>Column Header in .CSV Load File</th>
<th>Planning Property</th>
<th>Value</th>
<th>Default</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimension</td>
<td>Name of base dimension for which UDA is defined</td>
<td>Text, name of dimension for which the UDA will be defined; UDAs cannot be defined for Attribute dimensions</td>
<td>None</td>
<td>Yes</td>
</tr>
<tr>
<td>UDA</td>
<td>UDA being defined</td>
<td>Text, subject to Planning member naming restrictions</td>
<td>None</td>
<td>Yes</td>
</tr>
</tbody>
</table>

### Exchange Rate Dimension Properties

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

### Table 39  Exchange Rate Dimension Properties

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

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

<table>
<thead>
<tr>
<th>Column Header in CSV Load File</th>
<th>Planning Property</th>
<th>Value</th>
<th>Default</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary Member</td>
<td>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.</td>
<td>Text</td>
<td>None</td>
<td>Yes</td>
</tr>
<tr>
<td>Primary Enabled</td>
<td>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.</td>
<td>True, False, or an integer: non-zero is true; zero is false</td>
<td>None</td>
<td>No</td>
</tr>
<tr>
<td>Secondary Dimension</td>
<td>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.</td>
<td>Text</td>
<td>None</td>
<td>No</td>
</tr>
<tr>
<td>Secondary Parent</td>
<td>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.</td>
<td>Text</td>
<td>None</td>
<td>No</td>
</tr>
<tr>
<td>Relative Generation</td>
<td>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.</td>
<td>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.</td>
<td>None</td>
<td>No</td>
</tr>
<tr>
<td>Column Header in CSV Load File</td>
<td>Planning Property</td>
<td>Value</td>
<td>Default</td>
<td>Required</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>----------------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------</td>
<td>---------</td>
<td>----------</td>
</tr>
<tr>
<td>Auto Include</td>
<td>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.</td>
<td>True, False, or an integer: non-zero is true; zero is false</td>
<td>None</td>
<td>No</td>
</tr>
<tr>
<td>Include</td>
<td>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.</td>
<td>True, False, or an integer: non-zero is true; zero is false</td>
<td>True</td>
<td>No</td>
</tr>
<tr>
<td>Secondary Member</td>
<td>Secondary dimension member name that was enabled for Approvals. The Include, Owner, Reviewers, and Notifiees properties are dependent on this property. 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.</td>
<td>Text</td>
<td>None</td>
<td>Yes, if these column headers are defined: Owner, Reviewers, Notifiees</td>
</tr>
<tr>
<td>Owner</td>
<td>User name of the owner for the specified member. This property corresponds to the Owner column of the Assign Owners tab of the Planning Unit Hierarchy page.</td>
<td>Text</td>
<td>None</td>
<td>No</td>
</tr>
<tr>
<td>Reviewers</td>
<td>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. <strong>Note:</strong> Specify 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 user promotes the planning unit, the second reviewer in the list becomes the planning unit owner, and so on through the list of reviewers you create.</td>
<td>Text</td>
<td>None</td>
<td>No</td>
</tr>
<tr>
<td>Notifiees</td>
<td>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.</td>
<td>Text</td>
<td>None</td>
<td>No</td>
</tr>
</tbody>
</table>

**Smart List Dimension Properties**

SmartList Name, Operation, Label, Display Order, Missing Label, Use Form Setting, Entry ID, Entry Name, Entry Label
### Table 41  Smart List Dimension Properties

<table>
<thead>
<tr>
<th>Column Header in .CSV Load File</th>
<th>Planning Property</th>
<th>Value</th>
<th>Default</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>SmartList Name</td>
<td>Name of the Smart List</td>
<td>Text, name of the Smart List/enumeration</td>
<td>None</td>
<td>Yes</td>
</tr>
<tr>
<td>Operation</td>
<td>Operation</td>
<td>Text, addsmartlist—creates a new Smart List; addEntry—adds an entry to the Smart List</td>
<td>None</td>
<td>Yes</td>
</tr>
<tr>
<td>Label</td>
<td>Label field of Smart List</td>
<td>Text, label field of Smart List</td>
<td>Empty</td>
<td>No</td>
</tr>
<tr>
<td>Display Order</td>
<td>Display Order</td>
<td>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</td>
<td>ID</td>
<td>No</td>
</tr>
<tr>
<td>Missing Label</td>
<td>#Missing Drop Down label</td>
<td>Text, Missing Drop Down label</td>
<td>LABEL_NONE</td>
<td>No</td>
</tr>
<tr>
<td>Use Form Setting</td>
<td>#Missing Form</td>
<td>True—Form Setting; False—Drop Down Setting</td>
<td>Form Setting</td>
<td>No</td>
</tr>
<tr>
<td>Entry ID</td>
<td>Smartlist entry ID</td>
<td>ID for the Smart List entry</td>
<td>Default entry ID</td>
<td>No</td>
</tr>
<tr>
<td>Entry Name</td>
<td>Entry Name</td>
<td>N/A</td>
<td>None</td>
<td>Yes</td>
</tr>
<tr>
<td>Entry Label</td>
<td>Entry Label</td>
<td>N/A</td>
<td>None</td>
<td>Yes</td>
</tr>
</tbody>
</table>

---

**Working with Planning Outline Load**

**Subtopics**

- Running Planning Outline Load
- Importing from a File
- Exporting Metadata to a File
- Exporting Data to a File

Use Planning Outline Load 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 Outline Load.

**Note:** Planning Outline Load 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” on page 95.
Running Planning Outline Load

To import or export data or metadata using Planning Outline Load:

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” on page 95.

3. If importing, generate a load file for each dimension or set of data that you want to load. See “Generating Load Files” on page 97.

4. Log in to the Planning application.

5. Select Administration, then Outline Load, and then select a task:
   - To import data or metadata from a flat file, select Import from File. For option descriptions, see “Importing from a File” on page 141.
   - To export metadata to a flat file, select Export Metadata to File. For option descriptions, see “Exporting Metadata to a File” on page 143.
   - To export data to a flat file, select Export Data to File. For option descriptions, see “Exporting Data to a File” on page 143.

6. Click Run.

Importing 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 the Dimension Type field.

To import data, the load dimension must be selected in the Dimension drop down.

In Import from FlatFile, select the following options:

<table>
<thead>
<tr>
<th>Table 42 Import from File Options</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Option</strong></td>
</tr>
<tr>
<td>Dimension</td>
</tr>
<tr>
<td>Metadata File</td>
</tr>
<tr>
<td>Option</td>
</tr>
<tr>
<td>-------------------------------</td>
</tr>
<tr>
<td>Create Security Filters</td>
</tr>
<tr>
<td>Cube Refresh</td>
</tr>
<tr>
<td>Delete All Planning Units</td>
</tr>
<tr>
<td>Delete All Load Dimension Members</td>
</tr>
<tr>
<td>Note: Use caution. This option removes attribute bindings and approvals states.</td>
</tr>
<tr>
<td>Inherit When Plan Type Unspecified</td>
</tr>
<tr>
<td>Lock Load Dimension before Load</td>
</tr>
<tr>
<td>Maintain Order</td>
</tr>
<tr>
<td>Order Input Records</td>
</tr>
<tr>
<td>Dry Run</td>
</tr>
<tr>
<td>Date Format</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Driver Member Specification</td>
</tr>
<tr>
<td>Note: If no values are entered for this field but load and driver dimensions are set in the Administration/Data Load Settings window, then those settings will be used for data import. Values added to this field will override the values in the Data Load Settings window.</td>
</tr>
<tr>
<td>Clear Essbase Data</td>
</tr>
<tr>
<td>Note: If a data value exists in a cell and the imported data does not have value for that cell, the existing data will not be 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. For example:</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
Exporting Metadata to a File

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

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimension</td>
<td>The dimension you are exporting</td>
</tr>
</tbody>
</table>

After clicking **Run**, the browser will prompt you to either open the file or save it to your local machine.

Exporting Data to a File

Describe the Planning form that contains the data being exported. You must enter the plan type, row, column, and POV information that was entered during Planning 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>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plan Type</td>
<td>The plan type that was entered during Planning form design</td>
</tr>
<tr>
<td>Row Members</td>
<td>The row information that was entered during Planning form design Example: Idescendants(Ent_Total)</td>
</tr>
<tr>
<td>Column Members</td>
<td>The column information that was entered during Planning form design Example: Idescendants(YearTotal)</td>
</tr>
<tr>
<td>POV Members</td>
<td>The Page dimensions and member information Example: FY13,Current, BU Version_1,account1,Local</td>
</tr>
</tbody>
</table>

Data Integrator Adapter for Planning

Use Oracle Data Integrator Adapter for Planning to connect and integrate Planning with any database through Oracle Data Integrator (ODI). The adapter provides a set of Oracle Data Integrator Knowledge Modules (KMs) for loading metadata and data into Planning applications. For information on using ODI, see the *Oracle Hyperion Data Integration Management Adapter for Planning User’s Guide*. 
Using DIM Adapter for Planning

After installing and configuring Oracle Hyperion Data Integration Management Adapter for Planning, you can install and configure adapters to retrieve and write data for other Oracle products. After you configure an adapter, you must configure an application connection in Workflow Manager before extracting data from sources or writing data into targets. See the Oracle Data Integrator Adapter for Planning Online Help. Oracle Hyperion Data Integration Management Adapter for Planning is available only for Planning applications that use Planning application administration.

Enabling Data Load of Essbase Data

You specify parameters to enable data to be loaded directly into an 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 Chapter 7, “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.

Click Save.

For detailed instructions on loading, see the Planning adapter documentation described in “Overview” on page 93.

Loading from Administration Services

g. To load from Administration Services:

1 Open Administration Services Console.

2 See the 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 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.

<table>
<thead>
<tr>
<th>Entity</th>
<th>Category</th>
<th>Scenario</th>
<th>Month</th>
<th>Year</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>EUROPE</td>
<td>GROSS SALES</td>
<td>ACTUAL</td>
<td>JAN</td>
<td>2008</td>
<td>150</td>
</tr>
<tr>
<td>EUROPE</td>
<td>GROSS SALES</td>
<td>ACTUAL</td>
<td>FEB</td>
<td>2008</td>
<td>110</td>
</tr>
<tr>
<td>EUROPE</td>
<td>GROSS SALES</td>
<td>ACTUAL</td>
<td>MAR</td>
<td>2008</td>
<td>200</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Scenario</th>
<th>ACTUAL</th>
<th>FINAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>EUROPE</td>
<td>GROSS SALES</td>
<td>JAN</td>
</tr>
<tr>
<td>ASIA</td>
<td>NET INCOME</td>
<td>FEB</td>
</tr>
<tr>
<td>EUROPE</td>
<td>NET INCOME</td>
<td>FEB</td>
</tr>
<tr>
<td>ASIA</td>
<td>GROSS SALES</td>
<td>JAN</td>
</tr>
</tbody>
</table>

Loading with Performance Management Architect

You can load data and metadata with these Performance Management Architect features: flat files, interface tables, and Data Synchronization. For information, see the Oracle Hyperion Enterprise Performance Management Architect Administrator’s Guide.

Loading with FDM

You can load data with Oracle Hyperion Financial Data Quality Management. For information, see the Oracle Hyperion Financial Data Quality Management Administrator’s Guide.
About Forms

Subtopics

- Forms and Plan Types
- Forms and Access Permissions
- Forms and Currencies
- Forms and Versions
- Forms and Attributes
- Forms and Shared Members
- Forms and Calculations

Forms are grids for entering data. You can create many forms to meet your needs.

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” on page 150.

**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.

Form Components

Subtopics

- Point of View
- Page Axis
- Rows and Columns

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” on page 170 and “Managing User Variables” on page 192.

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” on page 170.

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” on page 168.

Designing Specific Types of Forms

Subtopics

- 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 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 FDM or FDMEE, 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 FDM or FDMEE, complete setup tasks, and load data or metadata to Planning. See the Oracle Hyperion Financial Data Quality Management Administrator’s Guide or Oracle Hyperion Financial Data Quality Management, Enterprise Edition Administrator’s Guide. You do not need to set properties or configure Planning for FDM.

- Give appropriate access permissions, for example, to the form and members, as described in Chapter 3, “Setting Up Access Permissions”. Drill-through is enabled against all dimensions that are sourced from Oracle Hyperion Financial Data Quality Management, Enterprise Edition. 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” on page 169. For information about creating formulas and using the mathematical functions, see Appendix C, “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” on page 166 and Chapter 8, “Managing Data Validation.”

**Designing Forms for Rolling Forecasts**

**Subtopics**

- About Rolling Forecasts
- Creating Rolling Forecasts
- Modifying Rolling Forecast Variables

**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 1  12-Monthly Rolling Forecast**

<table>
<thead>
<tr>
<th>Year and Period in Columns With No Additional Segment</th>
</tr>
</thead>
<tbody>
<tr>
<td>FY12</td>
</tr>
<tr>
<td>Jul</td>
</tr>
<tr>
<td>75</td>
</tr>
<tr>
<td>75</td>
</tr>
<tr>
<td>75</td>
</tr>
</tbody>
</table>

**Figure 2  Quarterly Rolling Forecast**

<table>
<thead>
<tr>
<th>FY12 Q2 Review</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
</tr>
</thead>
<tbody>
<tr>
<td>FY12 Q3 Review</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>FY12 Q4 Review</td>
<td>A</td>
<td>A</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>FY13 Q1 Review</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

**Figure 3  Quarterly Trailing Rolling Forecast (Rolling Quarters with a Cumulative Total)**

<table>
<thead>
<tr>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
<th>Q5</th>
<th>Q6</th>
<th>Q7</th>
<th>Q8</th>
<th>Q9</th>
<th>Q10</th>
<th>Q11</th>
<th>Q12</th>
<th>12 qtrs rolling</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project 1</td>
<td>25</td>
<td>35</td>
<td>45</td>
<td>55</td>
<td>65</td>
<td>75</td>
<td>85</td>
<td>95</td>
<td>105</td>
<td>115</td>
<td>125</td>
<td>135</td>
</tr>
<tr>
<td>Project 2</td>
<td>50</td>
<td>60</td>
<td>70</td>
<td>80</td>
<td>90</td>
<td>100</td>
<td>110</td>
<td>120</td>
<td>130</td>
<td>140</td>
<td>150</td>
<td>160</td>
</tr>
<tr>
<td>Project 3</td>
<td>75</td>
<td>85</td>
<td>95</td>
<td>105</td>
<td>115</td>
<td>125</td>
<td>135</td>
<td>145</td>
<td>155</td>
<td>165</td>
<td>175</td>
<td>185</td>
</tr>
<tr>
<td>Project 4</td>
<td>100</td>
<td>110</td>
<td>120</td>
<td>130</td>
<td>140</td>
<td>150</td>
<td>160</td>
<td>170</td>
<td>180</td>
<td>190</td>
<td>200</td>
<td>210</td>
</tr>
<tr>
<td>Project 5</td>
<td>125</td>
<td>135</td>
<td>145</td>
<td>155</td>
<td>165</td>
<td>175</td>
<td>185</td>
<td>195</td>
<td>205</td>
<td>215</td>
<td>225</td>
<td>235</td>
</tr>
</tbody>
</table>

**Figure 4  Rolling Forecast Where There are Additional Segments for Actual and Plan Year**

<table>
<thead>
<tr>
<th>Account</th>
<th>FY12</th>
<th>FY12</th>
<th>FY12</th>
<th>FY12</th>
<th>FY12</th>
<th>FY12</th>
<th>FY12</th>
<th>FY12</th>
<th>FY13</th>
<th>FY13</th>
<th>12 month Rolling Aug</th>
<th>Actual</th>
<th>Plan</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jul</td>
<td>Aug</td>
<td>Sep</td>
<td>Oct</td>
<td>Nov</td>
<td>Dec</td>
<td>Jan</td>
<td>Feb</td>
<td>Mar</td>
<td>Apr</td>
<td>May</td>
<td>Jun</td>
<td>Jul</td>
<td>Aug</td>
</tr>
<tr>
<td>50</td>
<td>75</td>
<td>75</td>
<td>75</td>
<td>75</td>
<td>75</td>
<td>75</td>
<td>75</td>
<td>75</td>
<td>75</td>
<td>75</td>
<td>75</td>
<td>125</td>
<td>125</td>
</tr>
</tbody>
</table>

**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” on page 159.

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:**
   
   - **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 this check box 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 the **Reuse existing substitution variables** check box, 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 the **Reuse existing substitution variables** check box, 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” on page 208.

- 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**

Subtopics

- Selecting and Opening Forms and Folders
- Previewing Forms
- Printing Form Definitions
- Searching for Forms
- Moving Forms and Ad Hoc Grids
- Deleting Forms
- Renaming Forms
- Importing and Exporting Forms
- Specifying How Many Forms Users Can Simultaneously Use

**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.
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” on page 375.

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 Planning directory points to the location `EPM_ORACLE_HOME\common\JRE\Sun\1.6.0\lib\fonts`. 

156  Managing Forms
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 \( \text{to search forward (down)} \) or \( \text{to search backwards (up)} \).

Moving Forms and Ad Hoc Grids

To move forms:
1. Select the form.
   See “Selecting and Opening Forms and Folders” on page 154.
   You can move multiple forms simultaneously if they are in the same folder.
2. Click **Move**.
3. Select the destination folder.
4. Click **OK**.

Deleting Forms

To delete forms:
1. Select the form.
   See “Selecting and Opening Forms and Folders” on page 154.
2. Click **Delete**.
3. Click **OK**.

Renaming Forms

To rename forms:
1. Select the form.
   See “Selecting and Opening Forms and Folders” on page 154.
2. Select **Rename**.
3. Enter the new name and click **OK**.
Importing and Exporting Forms

Administrators can use FormDefUtil.cmd (Windows) or FormDefUtil.sh (UNIX) to move form definitions between 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” on page 53.

**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
   ```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Purpose</th>
<th>Required?</th>
</tr>
</thead>
<tbody>
<tr>
<td>[-f:passwordFile]</td>
<td>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 passwordFile. See “Suppressing Password Prompts in Planning Utilities” on page 52.</td>
<td>No</td>
</tr>
<tr>
<td>import</td>
<td>export</td>
<td>Import or export the form definition.</td>
</tr>
<tr>
<td>filename</td>
<td>formname</td>
<td>-all</td>
</tr>
<tr>
<td>server name</td>
<td>Server name on which the Planning application resides.</td>
<td>Yes</td>
</tr>
<tr>
<td>user name</td>
<td>Administrator's name.</td>
<td>Yes</td>
</tr>
<tr>
<td>application</td>
<td>When used with export, the name of the Planning application containing the form definitions to export. When used with import, the name of the Planning application to which to import the form definition.</td>
<td>Yes</td>
</tr>
</tbody>
</table>

2. 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” on page 53.) You can copy the utility to any directory and launch it from there to save files to another directory.

Examples:

- To import one file:
To export one file:
FormDefUtil.cmd export Form1 localhost admin APP1

To export all form definitions:
```bash
FormDefUtil.cmd export -all localhost admin APP1
```

To import all form definitions:
```bash
FormDefUtil.cmd import -all localhost admin APP1
```

### 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 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.

### Creating Simple Forms

To create simple forms, define:
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” on page 147.
5. Optional: Provide instructions for working with the form.
6. Click Next to specify the form layout. See “Setting Form Layout” on page 160.

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.

Use the Layout tab to:

- Add form rows and columns
- Assign dimensions to columns and rows
- Select dimension members for users to work with (see Chapter 7, “Using the Member Selector”)
- Set grid properties for the form (see “Setting Form Grid Properties” on page 162)
- Set dimension properties (see “Setting Dimension Properties” on page 163)
Add formula rows and columns (see “Adding Formula Rows and Columns” on page 169)

Set display properties for the form (see “Setting Display Properties” on page 164)

Set printing options for the form (see “Setting Printing Options” on page 165)

Add and update validation rules in forms (see “Including Data Validation Rules in Forms” on page 166)

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” on page 154.

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 Chapter 7, “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>
<thead>
<tr>
<th>Table 45 Segment Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Option</strong></td>
</tr>
<tr>
<td>-----------------------------</td>
</tr>
<tr>
<td><strong>Apply to all rows</strong></td>
</tr>
<tr>
<td><strong>Apply to all columns</strong></td>
</tr>
<tr>
<td>Option</td>
</tr>
<tr>
<td>------------------------</td>
</tr>
<tr>
<td>Hide</td>
</tr>
<tr>
<td>Read-only</td>
</tr>
<tr>
<td>Show separator</td>
</tr>
<tr>
<td>Suppress hierarchy</td>
</tr>
<tr>
<td>Suppress missing data</td>
</tr>
</tbody>
</table>

**Column width**
- 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 column headings to fit in the displayed space
- Custom: Select a custom size to display more than 13 decimal places, up to 999 places

**Row height**
- Default: Use the row height defined at the grid level (under **Grid Properties**)
- Medium: Display standard row height
- Size-to-Fit: Force all row headings to fit in the displayed space
- Custom: Select a custom size in pixels for the row height

---

Optional: Add formula rows or columns. See “Adding Formula Rows and Columns” on page 169.

Optional: Add or update data validation rules. See “Including Data Validation Rules in Forms” on page 166.

### Setting Form Grid Properties

Form grid properties set the general form row and column display.

1. **To set form grid properties:**
   1. Open the form, and then click **Layout**.
      
      See “Selecting and Opening Forms and Folders” on page 154.
   2. In **Grid Properties**, set general row and column properties using the information in this table:

#### Table 46  Form Grid Properties

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Suppress missing blocks</td>
<td>(Rows only) Improves the performance of the <strong>Suppress missing data</strong> setting when suppressing a large number of rows, for example, 90% or more. The <strong>Suppress missing blocks</strong> 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. 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.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Suppress missing data</td>
<td>Hides rows or columns without data. Clear to display rows or columns with &quot;#MISSING&quot; in cells when data is missing.</td>
</tr>
</tbody>
</table>
| Default row height     | ● Medium  
● Size-to-Fit: Force all row headings to fit in the displayed space  
● Custom: Select a custom size in pixels for the row height                        |
| Default column width   | ● Small: Display seven decimal places  
● Medium: Display 10 decimal places  
● Large: Display 13 decimal places  
● Size-to-Fit: Force all column headings to fit in the displayed space  
● Custom: Select a custom size to display more than 13 decimal places, up to 999 places |

3 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:

1 Open the form, and then click **Layout**.

   See “Selecting and Opening Forms and Folders” on page 154.

2 Click in a point of view, page, row, or column to set dimension properties.

3 Select **Dimension Properties**:

<table>
<thead>
<tr>
<th>Table 47</th>
<th>Dimension Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>Apply to all row dimensions</td>
<td>Applies properties to all row dimensions</td>
</tr>
<tr>
<td>Apply to all column dimensions</td>
<td>Applies properties to all column dimensions</td>
</tr>
<tr>
<td>Apply to all page dimensions</td>
<td>Applies properties to all page dimensions</td>
</tr>
<tr>
<td>Apply to all POV dimensions</td>
<td>Applies properties to all point of view dimensions</td>
</tr>
<tr>
<td>Member Name</td>
<td>Displays the member name</td>
</tr>
<tr>
<td>Alias</td>
<td>Displays the member alias</td>
</tr>
<tr>
<td>Member Formula</td>
<td>Displays member formulas</td>
</tr>
<tr>
<td>Hide dimension</td>
<td>Hides the dimension</td>
</tr>
<tr>
<td>Show consolidation operators</td>
<td>Displays consolidation operators</td>
</tr>
</tbody>
</table>
### Property Description

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start expanded</td>
<td>Available only for dimensions on row or columns, choosing this option initially displays the dimension member list expanded.</td>
</tr>
<tr>
<td>Enable custom attributes</td>
<td>Available only for dimensions on row or columns, enables custom attributes.</td>
</tr>
</tbody>
</table>

4. 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:

1. **Open the form, and then click Layout.**
   
   See “Selecting and Opening Forms and Folders” on page 154.

2. **Select Display Properties, and then select form options:**

   - **Make form read-only**: You cannot set this option for composite forms.
   - **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 *Oracle Hyperion Planning User’s Guide*.

   - **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 *Oracle Hyperion Planning User’s Guide*. 

---

164 **Managing Forms**
Enable Grid Spread

For information about using Grid Spread, see Oracle Hyperion Planning User’s Guide.

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 Oracle Hyperion Planning User’s Guide, Chapter 3, “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, There are no valid rows of data for this form.

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” on page 154.

2 Select Printing Options, and then set preferences for printing form information:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Include supporting detail</td>
<td>Include supporting detail as extra rows in PDF files. Specify display format:</td>
</tr>
<tr>
<td></td>
<td>Normal Order: Prints supporting detail in the same order as on the Supporting Detail page, after the member it is associated with</td>
</tr>
<tr>
<td></td>
<td>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</td>
</tr>
<tr>
<td>Show comments</td>
<td>Display text notes associated with cells</td>
</tr>
<tr>
<td>Format data</td>
<td>Apply number format settings from the form to the displayed data</td>
</tr>
<tr>
<td>Show attribute members</td>
<td>If attribute members are selected in the form, display them in PDF files</td>
</tr>
<tr>
<td>Apply precision</td>
<td>Apply form precision settings (desired number of decimal points) to the displayed data in PDF files</td>
</tr>
<tr>
<td>Show currency codes</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>Show account annotations</td>
<td>If account annotations are enabled for the form, select to display account annotations in PDF files</td>
</tr>
</tbody>
</table>
You can also create reports for forms, as described in “Customizing Reports” on page 375.

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 Chapter 8, “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” on page 154.

2 Select Validation Rules, and then select an option:

   - **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, 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” on page 211.

4 In the form, click Next to continue building the form, and then validate and save the form.
Setting 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” on page 154.

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>
<thead>
<tr>
<th>Value</th>
<th>Minimum Precision</th>
<th>Maximum Precision</th>
<th>Displayed Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0</td>
<td>Any</td>
<td>100</td>
</tr>
<tr>
<td>100</td>
<td>3</td>
<td>Any number greater than or equal to 3 or None</td>
<td>100.000</td>
</tr>
<tr>
<td>100.12345</td>
<td>Any number less than or equal to 5</td>
<td>None</td>
<td>100.12345</td>
</tr>
<tr>
<td>100.12345</td>
<td>7</td>
<td>None</td>
<td>100.1234500</td>
</tr>
<tr>
<td>100.12345</td>
<td>Any number less than or equal to 3</td>
<td>3</td>
<td>100.123</td>
</tr>
<tr>
<td>100.12345</td>
<td>0</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>100.12345</td>
<td>2</td>
<td>4</td>
<td>100.1234</td>
</tr>
<tr>
<td>100</td>
<td>2</td>
<td>4</td>
<td>100.00</td>
</tr>
</tbody>
</table>

Notes:

- By default, the precisions settings you select here override the precision set for the currency member (see “Creating Currencies” on page 343). 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 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 99.99999999999998 when you click into Q1.

3 In **Smart View Option**, select **Enable offline usage** if users can work on the form offline.

This setting is applicable when the application is enabled for offline (the default setting). You can prevent the current application from being used offline by selecting Administration, then Application, Properties, then the Application Properties tab, and then changing **ENABLE_FOR_OFFLINE** to **False**. See “Setting Application and System Properties” on page 39.

**Note:** Smart View options are not available for composite forms.

4 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

5 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.

6 Select **Enable Dynamic User Variables** to allow dynamic user variables in the form (see Oracle Hyperion Planning User’s Guide).

7 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” on page 154.
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 Chapter 7, “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” on page 419).

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” on page 154.

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.
7 Define the formula to use for the row or column by entering the formula name in the Formula field, and then clicking Validate. See “Editing Formulas” on page 419.

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” on page 192.

➤ To define page axis and point of view:

1 Open the form, and then click Layout. See “Selecting and Opening Forms and Folders” on page 154.

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 Chapter 7, “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.

4 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 Planning sets the page selection to the most recently used selection.

5 In Dimension Properties, select or clear options for page dimensions. See “Setting Dimension Properties” on page 163.

6 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.

7 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 Chapter 7, “Using the Member Selector.”
In Dimension Properties, select or clear options for point of view dimensions. See “Setting Dimension Properties” on page 163.

Click Save to save your work and continue, or click Finish to save your work and close the form.

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.

To edit simple forms:

1. Select the form, then click Edit (see “Selecting and Opening Forms and Folders” on page 154).
2. Select:
   a. Properties to edit the form name, description and instructions. See “Creating Simple Forms” on page 159.
   b. Layout to edit form layout. See “Setting Form Layout” on page 160.
   c. Other Options to edit form precision and to change which context menus are associated with the form. See “Setting Other Options” on page 167.
   d. Business Rules to change which business rules are associated with the form, or modify business rule properties. See “Using Business Rules” on page 181.
3. Click Finish to save your work and close the form.

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” on page 154).

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:
   a. Properties to edit the composite form name, description or instructions. See “Creating Composite Forms” on page 172.
   b. Layout to edit form layout and properties. See “Setting Composite Form Layout” on page 172.
   c. Business Rules to change which business rules are associated with the form, or modify business rule properties. See “Using Business Rules” on page 181.
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” on page 154.
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” on page 172.
8. Set the composite form section properties. See “Setting Composite Form Section Properties” on page 175.
9. Set the composite form Point of View and Page display options. See “Setting Composite Form Point of View and Page Dimensions” on page 176.
10. Click Save to save your work and continue, or click Finish to save your work and close the form.

Setting Composite Form Layout

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” on page 154.
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” on page 174, “Rearranging Forms in the Layout” on page 174 and “Deleting Simple Forms from a Composite Form” on page 175.

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” on page 171. Access permissions apply as described in “Forms and Access Permissions” on page 148.
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” on page 148.

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” on page 171.
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 **X**.
- 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” on page 154.
2. **Expand Section Properties**.
3. **Click in a composite form section and set the properties as desired**.

<table>
<thead>
<tr>
<th>Table 50</th>
<th>Composite Form Section Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td><strong>Forms</strong></td>
<td>Displays the simple forms in the section. The following options are available for each form selected:</td>
</tr>
<tr>
<td></td>
<td>• Display forms as tabs</td>
</tr>
<tr>
<td></td>
<td>• Add form</td>
</tr>
<tr>
<td></td>
<td>• Remove form</td>
</tr>
<tr>
<td></td>
<td>• Edit form label</td>
</tr>
<tr>
<td></td>
<td>• Move to top</td>
</tr>
<tr>
<td></td>
<td>• Move up</td>
</tr>
<tr>
<td></td>
<td>• Move down</td>
</tr>
<tr>
<td></td>
<td>• Move to bottom</td>
</tr>
<tr>
<td><strong>Name</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Height</strong></td>
<td>Section height. Select:</td>
</tr>
<tr>
<td></td>
<td>• Automatic to have Planning set the height.</td>
</tr>
<tr>
<td></td>
<td>• % (percentage sign) to set section height to a percentage of the composite form height.</td>
</tr>
<tr>
<td><strong>Width</strong></td>
<td>Width of the section. Select:</td>
</tr>
<tr>
<td></td>
<td>• Automatic to have Planning set the width.</td>
</tr>
<tr>
<td></td>
<td>• % (percentage sign) to set section width to a percentage of the composite form width.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Forms per Row</strong></td>
<td>Select:</td>
</tr>
<tr>
<td>• <strong>Automatic</strong></td>
<td>to have Planning set the number.</td>
</tr>
<tr>
<td>• <strong>Select a number from 1 to 20</strong></td>
<td></td>
</tr>
<tr>
<td>The default is one form per row. If <strong>Forms per Column</strong> is set to a value other than <strong>Automatic</strong>, <strong>Forms per Row</strong> is set to <strong>Automatic</strong>.</td>
<td></td>
</tr>
<tr>
<td><strong>Note:</strong> If you have grouped the forms as tabs, this option is not available.</td>
<td></td>
</tr>
<tr>
<td><strong>Forms per Column</strong></td>
<td>Select:</td>
</tr>
<tr>
<td>• <strong>Automatic</strong></td>
<td>to have Planning set the number.</td>
</tr>
<tr>
<td>• <strong>Select a number from 1 to 20</strong></td>
<td></td>
</tr>
<tr>
<td>The default is one form per column. If <strong>Forms per Row</strong> is set to a value other than <strong>Automatic</strong>, <strong>Forms per Column</strong> is set to <strong>Automatic</strong>.</td>
<td></td>
</tr>
<tr>
<td><strong>Note:</strong> If you have grouped the forms as tabs, this option is not available.</td>
<td></td>
</tr>
<tr>
<td><strong>Set scope for all common dimensions as global</strong></td>
<td>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.</td>
</tr>
</tbody>
</table>

### 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 5  Master Composite Form: “New Computers”**

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.
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” on page 154.

2. **Right-click the form, and select Tag as Master Composite Form.**

   ![Tag as Master Composite Form](image)

   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” on page 154.
Click a composite form section, and then right-click on a form.

Select Display as Chart.

Display as Chart toggles with Display as Grid, allowing you to switch between them.

On Chart Properties, select a chart type:

- **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 how advice on selecting a chart type.

Click OK.

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.

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.
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 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

With appropriate access, users can launch business rules from 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” on page 183.

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 Chapter 7, “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 calc 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” on page 182.

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.

1. Click the Business Rules tab.
2. 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.
3. 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.
4. 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” on page 184.
5. 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
6. 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 you enabled members that are used on forms and in runtime prompts for dynamic children, users can create additional members if the members that they enter in runtime prompts do not exist. See “About Dynamic Members” on page 326.

When launching business rules with runtime prompts, 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” on page 182. To understand how other settings and conditions affect runtime prompts, see “Understanding Runtime Prompts” on page 184.
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” on page 39.

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” on page 182)

- 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

If you enable members that are used on forms and in runtime prompts for dynamic children, users can define additional members if the members that they enter in runtime prompts do not exist.

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 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 Point/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>
<thead>
<tr>
<th>Availability of Override Value and member on the Page/Point of View</th>
<th>Use Members on Form option is selected</th>
<th>Hide Runtime Prompt property is set during runtime prompt design</th>
<th>Hide Prompt option is selected for the form</th>
<th>Result on Runtime Prompt</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use as Override Value</strong> is set and <strong>Override Value</strong> is available or the member is available on the Page/Point of View to use as the runtime prompt value.</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes or No Setting is ignored</td>
<td>The business rule runs without displaying the runtime prompt to users. Instead, the runtime prompt value is taken from the <strong>Override Value</strong> or Page/Point of View member.</td>
</tr>
<tr>
<td><strong>Use as Override Value</strong> is set and <strong>Override Value</strong> is available or the member is available on the Page/Point of View to use as the runtime prompt value.</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>If all runtime prompts can be pre-filled from the <strong>Override Value</strong> 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 <strong>Override Value</strong> or Page/Point of View context, then all runtime prompts display, with values pre-filled wherever possible. All others follow Principles 1 and 3.</td>
</tr>
<tr>
<td>Availability of Override Value and member on the Page/Point of View</td>
<td>Use Members on Form option is selected</td>
<td>Hide Runtime Prompt property is set during runtime prompt design</td>
<td>Hide Prompt option is selected for the form</td>
<td>Result on Runtime Prompt</td>
</tr>
<tr>
<td>---------------------------------------------------------------</td>
<td>--------------------------------------</td>
<td>-------------------------------------------------</td>
<td>---------------------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td><strong>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.</strong></td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>The runtime prompt is displayed to users, with values pre-filled from the <strong>Override Value</strong> or Page/Point of View.</td>
</tr>
<tr>
<td><strong>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.</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes or No Setting is ignored</td>
<td>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 <strong>Hide Prompt</strong> setting is ignored and the runtime prompt displayed.</td>
</tr>
<tr>
<td><strong>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.</strong></td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>The runtime prompt is displayed to users, with values pre-filled according to Principle 3.</td>
</tr>
<tr>
<td><strong>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.</strong></td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>If all runtime prompts can be pre-filled from the <strong>Override Value</strong> 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 <strong>Override Value</strong>, then all runtime prompts display, with values pre-filled wherever possible. All others follow Principles 1 and 3.</td>
</tr>
<tr>
<td><strong>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.</strong></td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>The runtime prompt is displayed to users, with values pre-filled according to Principle 3.</td>
</tr>
<tr>
<td><strong>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.</strong></td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>The runtime prompt is displayed to users, with values pre-filled according to Principles 1 and 3.</td>
</tr>
<tr>
<td><strong>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.</strong></td>
<td>No</td>
<td>Yes</td>
<td>Not available</td>
<td>The business rule runs without displaying the runtime prompt to users. Instead, the design-time values are used.</td>
</tr>
<tr>
<td>Availability of Override Value and member on the Page/Point of View</td>
<td>Use Members on Form option is selected</td>
<td>Hide Runtime Prompt property is set during runtime prompt design</td>
<td>Hide Prompt option is selected for the form</td>
<td>Result on Runtime Prompt</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td><strong>Use as Override Value</strong> is set and <strong>Override Value</strong> is available or the member is available on the Page/Point of View to use as the runtime prompt value.</td>
<td>No</td>
<td>No</td>
<td>Not available</td>
<td>The runtime prompt is displayed to users, with values pre-filled according to Principle 3.</td>
</tr>
<tr>
<td><strong>Use as Override Value</strong> is not set and the member is not available on the Page/Point of View to use as the runtime prompt value.</td>
<td>No</td>
<td>Yes</td>
<td>Not available</td>
<td>The business rule runs without displaying the runtime prompt to users. Instead, the design-time values are used.</td>
</tr>
<tr>
<td><strong>Use as Override Value</strong> is not set and the member is not available on the Page/Point of View to use as the runtime prompt value.</td>
<td>No</td>
<td>No</td>
<td>Not available</td>
<td>The runtime prompt is displayed to users, with values pre-filled according to Principle 3.</td>
</tr>
</tbody>
</table>

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 Calculation Manager, administrators can set runtime Security for a member or members:

- **Approvals**: 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” on page 188.

Designing Secure Runtime Prompts

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 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” on page 188.

Launching Business Rules With a Utility

Using the CalcMgrCmdLineLauncher.cmd utility, administrators can launch—from the Command Prompt—business rules or rulesets created with Calculation Manager.

Notes:

- Use the /Validate option to check the command syntax before you launch the business rule or ruleset.
- You cannot launch cross-application rulesets with this utility.
- 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.
- If the launched ruleset has a runtime prompt, default launch values are used, and you do not specify a runtime prompt values file.

To launch a business rule or ruleset with CalcMgrCmdLineLauncher.cmd:

1. When launching a business rule (not a ruleset) 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
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” on page 53). 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:


Table 52 CalcMgrCmdLineLauncher Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Purpose</th>
<th>Required?</th>
</tr>
</thead>
<tbody>
<tr>
<td>[-f:passwordFile]</td>
<td>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 passwordFile. See “Suppressing Password Prompts in Planning Utilities” on page 52.</td>
<td>No</td>
</tr>
<tr>
<td>/A:appname</td>
<td>Specify the Planning application from which the business rule is launched</td>
<td>Yes</td>
</tr>
<tr>
<td>/U:username</td>
<td>Specify the administrator’s user name</td>
<td>Yes</td>
</tr>
<tr>
<td>/D:database</td>
<td>Specify the name of the plan type against which the calculation is launched</td>
<td>Yes</td>
</tr>
<tr>
<td>[/R:business rule name]</td>
<td>Specify the name of the business rule or ruleset to be launched</td>
<td>Yes. Specify either a business rule or a ruleset, but not both.</td>
</tr>
<tr>
<td>[/S:business ruleset name]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>/F:runtime prompts file</td>
<td>Specify the name of the file containing business rule’s runtime prompt names and values</td>
<td>Yes, when launching a business rule having a runtime prompt. When launching a ruleset, default launch values are used. If you specify a runtime prompts file when launching a ruleset, the runtime prompts file is ignored.</td>
</tr>
<tr>
<td>[/validate]</td>
<td>Checks the command syntax only; does not launch the business rule or ruleset. Any errors are displayed on the console and written to the Calculation Manager log file. If the CalcMgrLog4j.properties file is in the Classpath, the log file is generated in the EPM_ORACLE_INSTANCE/diagnostics/logs/planning directory. Each rule in a ruleset is validated.</td>
<td>No</td>
</tr>
<tr>
<td>/?</td>
<td>Print the syntax and options for CalcMgrCmdLineLauncher.cmd</td>
<td>No</td>
</tr>
</tbody>
</table>
For example to launch the rule named Depreciate, using runtime prompt values in a file called Values.txt, enter:

```
```

3 If prompted, enter your password.

**Customizing Error Messages**

Administrators can use the 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")), ERROR)
    ```

For instructions on customizing the HspCustomMsgs file, see “Customizing Text, Color, and Images” on page 384.

- You cannot use the function in member formulas.

For more information on the @RETURN function, see the *Oracle Essbase Technical Reference*. 
Working with User Variables

Subtopics

- About User Variables
- Managing User Variables
- Creating User Variables
- Deleting 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. After that, 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. After that, 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” on page 209.

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" on page 170.

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” on page 193.
3. When designing the form, associate the user variable with the form.
   
   See “About Selecting User Variables as Members” on page 209.
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” on page 209. After that, 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.

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 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 Administration Services Console or ESSCMD.
For more information about selecting substitution variables, see “About Selecting Substitution Variables as Members” on page 208.

Creating and Assigning Values to Substitution Variables Using Planning

To create and assign values to substitution variables using 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 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.

Importing Form Definitions

Use the ImportFormDefinition utility to import a form definition, but not data, from a text-based file into a 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:
1. Prepare the form for importing.
   See “Preparing Forms” on page 195.
2. Prepare the data file.
   See “Preparing Data Files” on page 195.
3. Run the utility.
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” on page 197.

Preparing Data Files

ImportFormDefinition imports data from a text-based file in 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” on page 53.
To launch `ImportFormDefinition`:

1. From the `planning1` directory, enter the command using this syntax:

   ```
   ```

   **Table 53  ImportFormDefinition Utility Parameters**

<table>
<thead>
<tr>
<th>Setting</th>
<th>Purpose</th>
<th>Required?</th>
</tr>
</thead>
<tbody>
<tr>
<td>[-f:passwordFile]</td>
<td>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” on page 52.</td>
<td>No</td>
</tr>
<tr>
<td>/A</td>
<td>Application name.</td>
<td>Yes</td>
</tr>
<tr>
<td>/U</td>
<td>Administrator user name.</td>
<td>Yes</td>
</tr>
<tr>
<td>/F</td>
<td>Form name.</td>
<td>Yes</td>
</tr>
<tr>
<td>/D</td>
<td>Name and location of the 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.</td>
<td>Yes</td>
</tr>
<tr>
<td>/AR</td>
<td>Add rows from the data file (on by default). Disable by specifying /-AR. For example, you can define rows in Planning and import only column definitions.</td>
<td>No</td>
</tr>
<tr>
<td>/AC</td>
<td>Add columns from the data file (on by default). Disable by specifying /-AC.</td>
<td>No</td>
</tr>
<tr>
<td>/KC</td>
<td>Keep member selections for columns in the form (on by default). Clear columns by specifying /-KC. 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.</td>
<td>No</td>
</tr>
<tr>
<td>/KR</td>
<td>Keep member selections for rows in the form (on by default). Clear member selections in rows by specifying /-KR. 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.</td>
<td>No</td>
</tr>
<tr>
<td>/SR</td>
<td>Sort rows in the form (on by default). Disable by specifying /-SR.</td>
<td>No</td>
</tr>
<tr>
<td>/SC</td>
<td>Sort columns in the form (on by default). Disable by specifying /-SC.</td>
<td>No</td>
</tr>
</tbody>
</table>

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 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 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 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.

**Planning Offline Considerations**

**Subtopics**

- About Offline Planning
- Offline Form Considerations
- Offline User Considerations
- Business Rule Considerations for Offline Calculations

**About Offline Planning**

Offline Planning allows users to take forms offline and work while disconnected from the server. Note these important considerations.

- For information about installing and configuring offline Planning, see *Oracle Enterprise Performance Management System Installation and Configuration Guide*.

- For information about using the Install Smart View menu item available from the `Tools` menu, see the *Oracle Hyperion Enterprise Performance Management Workspace User’s Guide*.
Offline Form Considerations

- Forms used offline should be self-contained.

  To ensure that values calculated offline are correct, all necessary dependencies pertinent to forms taken offline must be available offline. All members, forms, member formulas, business rules, and dynamic calculations on which a form depends for accurate results offline must also be taken offline.

- Data not downloaded from the server is not available offline.

- Dependent data required for business rules to run offline must be taken offline.

Offline User Considerations

Considerations for users working on offline forms:

- Although offline users can select Sync Back To Server, offline users must have write permission for a cell to save the changed cell value to the server.

- If a form member is deleted on the server while a user is working with that form and member offline, the offline user’s changes to that member are not saved to the server when Sync Back To Server is selected.

- If more than one user modifies the same data, the last values synchronized back are saved. To prevent data loss or miscalculations, use planning units and approvals on the server to control data access.

- Offline users can enter data in multiple currencies just as online users can. However, currency conversion is currently not supported when working offline. When users change currencies offline, values are not recalculated and displayed in the new currency.

- Leave the form property Enable Offline Usage set when a form is used offline, which allows users to save data changed offline back to the server.

- Use planning units to prevent two users from working with a form simultaneously. An offline user can lose access to members taken offline if an online user working with the same form clears the values in a row or column that has Suppress Missing Data set.

- Smart View does not currently support hidden forms; they are not downloaded when users take forms offline that are associated with hidden forms.

- Smart View does not currently support composite forms; they are not listed when users select forms to take offline.

Business Rule Considerations for Offline Calculations

Considerations for running business rules offline:

- Business rule calculations can use only data and objects taken offline; rules do not have access to data or objects stored on the server.

- Runtime prompts are not supported for business rules with the Run on Load property set.
- Business rules set to **Run on Save** that require user input for a runtime prompt are not supported when synchronizing back to the server.
- The **Run on Load** and **Run on Save** options can use only business rules and dependent data available offline.
About Working With Members

Use the Member Selection dialog box to select members and other information for 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 Planning features. For example, you can select members for form management, business rule runtime prompts, and Clear Cell Details. If variables and attributes are defined, you can also select variables and attributes. 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 one or more 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” on page 204.

b. Click 📚.

c. **Optional:** To go to the next or previous item, click 🔼 or 🔻.

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**. The 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.
Move members to or from the **Selected Members** list:

- Click ➔ to add selected members.
- Click ← to remove selected members.
- Click ← to remove all members.
- Click ➔ to add all members.
- Click in the middle of the dialog box to select a range of members based on hierarchy relationships.

<table>
<thead>
<tr>
<th>Table 54 Member Relationships</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Relationship</strong></td>
</tr>
<tr>
<td>Member</td>
</tr>
<tr>
<td>Descendants</td>
</tr>
<tr>
<td>Descendants (inc)</td>
</tr>
<tr>
<td>Ancestors</td>
</tr>
<tr>
<td>Ancestors (inc)</td>
</tr>
<tr>
<td>Siblings</td>
</tr>
<tr>
<td>Siblings (inc)</td>
</tr>
<tr>
<td>Parents</td>
</tr>
<tr>
<td>Parents (inc)</td>
</tr>
<tr>
<td>Children</td>
</tr>
<tr>
<td>Children (inc)</td>
</tr>
<tr>
<td>Level 0 Descendants</td>
</tr>
</tbody>
</table>

**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.

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” on page 209.
- **Substitution Variables:** Select members as described in “About Selecting Substitution Variables as Members” on page 208. The selection depends on the preference setting for **Enable the Display of Substitution Variables**, described in “Specifying System Settings” on page 277.
- **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” on page 205.

Optional: To select functions that apply to custom attributes, click 📌. To filter by dimension, select an option from the Dimension drop-down list.

### Table 55  Attribute Value Selection

<table>
<thead>
<tr>
<th>Operator</th>
<th>Attribute Values Included</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equal</td>
<td>Equal to the selected attribute</td>
</tr>
<tr>
<td>NotEqual</td>
<td>Not equal to the selected attribute</td>
</tr>
<tr>
<td>Greater</td>
<td>Greater than the selected attribute</td>
</tr>
<tr>
<td>GreaterOrEqual</td>
<td>Greater than or equal to the selected attribute</td>
</tr>
<tr>
<td>Less</td>
<td>Less than the selected attribute</td>
</tr>
<tr>
<td>LessOrEqual</td>
<td>Less than or equal to the selected attribute</td>
</tr>
</tbody>
</table>

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 56  Wildcard Characters

<table>
<thead>
<tr>
<th>Wildcard</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>?</td>
<td>Match any single character</td>
</tr>
<tr>
<td>*</td>
<td>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 “<strong>cash</strong>” and returns “Restricted Cash”, “Cash Equivalents”, “Cash”, and “Noncash Expenses” because the word “cash” appears within each matched item.</td>
</tr>
<tr>
<td>#</td>
<td>Match any single number (0-9)</td>
</tr>
<tr>
<td>[list]</td>
<td>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 [10-9]. To use the “-” character as part of the list, enter it at the beginning of the list. For example, [-@&amp;] uses the characters within the brackets as wildcard characters.</td>
</tr>
<tr>
<td>[list]</td>
<td>Match any single character not found within a specified list of characters. The “-” character can also be used to indicate a range, such as [1A-Z] or [10-9].</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Selected Operator</th>
<th>Selected Attribute Value</th>
<th>Result</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equal</td>
<td>Large</td>
<td>5, 6</td>
<td>The Equal operator is applied to all level 0 descendants of Large, which includes 5 and 6.</td>
</tr>
<tr>
<td>Less</td>
<td>Medium</td>
<td>1, 2, 3</td>
<td>The Less operator is applied to all level 0 descendants of Medium. This includes values &lt; 3 OR &lt; 4, which results in 1, 2, and 3.</td>
</tr>
<tr>
<td>Greater</td>
<td>Medium</td>
<td>4, 5, 6</td>
<td>The Greater operator is applied to all level 0 descendants of Medium. This includes values &gt; 3 OR &gt; 4, which results in 4, 5, and 6.</td>
</tr>
<tr>
<td>GreaterOrEqual</td>
<td>Medium</td>
<td>3, 4, 5, 6</td>
<td>The GreaterOrEqual operator is applied to all level 0 descendants of Medium. This includes values &gt;=3 OR &gt;=4, which results in 3, 4, 5, and 6.</td>
</tr>
<tr>
<td>LessOrEqual</td>
<td>Medium</td>
<td>1, 2, 3, 4</td>
<td>The LessOrEqual operator is applied to all level 0 descendants of Medium. This includes values &lt;=3 OR &lt;=4, which results in 1, 2, 3, and 4.</td>
</tr>
<tr>
<td>Selected Operator</td>
<td>Selected Attribute Value</td>
<td>Result</td>
<td>Explanation</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------------------</td>
<td>--------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>NotEqual</td>
<td>Medium</td>
<td>1, 2, 5, 6</td>
<td>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.</td>
</tr>
</tbody>
</table>

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 58  Example: Text Attribute Evaluation

<table>
<thead>
<tr>
<th>Selected Operator</th>
<th>Selected Attribute Value</th>
<th>Result</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equal</td>
<td>Medium</td>
<td>Box, Carton</td>
<td>The Equal operator is applied to all level 0 descendants of Medium, which includes Box and Carton.</td>
</tr>
<tr>
<td>NotEqual</td>
<td>Medium</td>
<td>Envelope, Packet, Barrel, Crate</td>
<td>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.</td>
</tr>
<tr>
<td>Less</td>
<td>Medium</td>
<td>Box, Packet, Envelope</td>
<td>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.</td>
</tr>
<tr>
<td>Selected Operator</td>
<td>Selected Attribute Value</td>
<td>Result</td>
<td>Explanation</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------------------</td>
<td>--------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>LessOrEqual</td>
<td>Medium</td>
<td>Envelope, Packet, Box, Carton</td>
<td>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.</td>
</tr>
</tbody>
</table>

### 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” on page 58.

- 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” on page 168.

- 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” on page 168.
- To set display, functionality, and printing options, see “Setting Other Options” on page 167.
- To set up reports that include forms or form definitions, see “Customizing Reports” on page 375.

### 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 Essbase server.

**Notes:**

- When you open or calculate values on forms, the 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” on page 193.

  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” on page 159).

2. On 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” on page 192.

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 planners select a variable for a form, they do so in preferences. After that, they can update the variable in preferences or in the form.

To select user variables for forms:

1. Create the form (see “Creating Simple Forms” on page 159).

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.
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” on page 229.

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 if 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 to a new location 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” on page 215.

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” on page 216.

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” on page 216.

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” on page 227.

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 one or more 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” on page 216.
- 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” on page 214.

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. Any errors found for the rule must be corrected before you can 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, 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 ▲, ▼, ≥, or ≤. See “Order of Evaluation and Execution for Data Validation Rules” on page 215.

- 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” on page 159.

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 does not display for a rule, ensure that the rule is valid, and that the rule permits cell processing instructions. For example, cell processing instructions are included for Else, Range, and Then conditions. The icon does not display until all required columns are selected for a rule. See “Creating and Updating Data Validation Rules” on page 211.

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, type 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” on page 254.

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” on page 211
- “Formatting Cells and Setting the Promotional Path” on page 214
- “Order of Evaluation and Execution for Data Validation Rules” on page 215
- “Conditions Supported by the Rule Builder” on page 216
- “Data Validation Conditional Operators” on page 227
- “Data Validation Rule Scenarios” on page 229

### 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, 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 59  Priority for Rules in Forms

<table>
<thead>
<tr>
<th>Default Condition Priority</th>
<th>Validation Message</th>
<th>Color</th>
<th>Do Not Promote</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (lowest)</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>3</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>5 (highest)</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

For additional information about data validation rules, see:

- “Creating and Updating Data Validation Rules” on page 211
- “Formatting Cells and Setting the Promotional Path” on page 214
- “Viewing Data Validation Rules” on page 215
- “Conditions Supported by the Rule Builder” on page 216
- “Data Validation Conditional Operators” on page 227
- “Data Validation Rule Scenarios” on page 229

**Conditions Supported by the Rule Builder**

**Subtopics**

- If Condition Values
- Then Condition Values
- Range Condition Values

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” on page 217.
- Then: “Then Condition Values” on page 225.
- Check Range, Range: “Range Condition Values” on page 226.

For additional information about data validation, see:
If Condition Values

Subtopics

- Current Cell Value
- Cell Value
- Column Value
- Row Value
- Cross Dim Member
- Member Name
- Member
- Account Type
- Version Type
- Variance Reporting Type
- UDA
- Attribute

These values are supported by the data validation rule builder for If conditions:

- “Current Cell Value” on page 218
- “Cell Value” on page 219
- “Column Value” on page 220
- “Row Value” on page 220
- “Cross Dim Member” on page 221
- “Member Name” on page 222
- “Member” on page 222
- “Account Type” on page 223
- “Version Type” on page 224
- “Variance Reporting Type” on page 224
- “UDA” on page 224
- “Attribute” on page 225
For information on other conditions, see “Conditions Supported by the Rule Builder” on page 216.

For additional information about data validation, see:
- “Creating and Updating Data Validation Rules” on page 211
- “Order of Evaluation and Execution for Data Validation Rules” on page 215
- “Formatting Cells and Setting the Promotional Path” on page 214
- “Viewing Data Validation Rules” on page 215
- “Conditions Supported by the Rule Builder” on page 216
- “Data Validation Conditional Operators” on page 227
- “Data Validation Rule Scenarios” on page 229

**Current Cell Value**

**Action:**
The action is performed only when the value in the current data cell on which the rule is invoked satisfies this condition.

**Operators:**
The operators available for this function are: =, ! =, <, <=, >, >=, 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:**

![Diagram of data validation rule builder](image)

**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.
**Cell Value**

**Action:**
The action is performed only when the value for the specified cell satisfies the condition.

**Operators:**
The operators available for this function are =, !=, <, <=, >, >=, 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:**

**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).
**Column Value**

**Action:**
The action is performed only when the value for the specified column satisfies the condition.

**Operators:**
The operators available for this function are =, !=, <, <=, >, >=, 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:**

<table>
<thead>
<tr>
<th>IF</th>
<th>Column Value</th>
<th>A</th>
<th>&lt;</th>
<th>Value</th>
<th>3</th>
</tr>
</thead>
</table>

**Condition Evaluation:**

A design time column can expand to one or more 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.

<table>
<thead>
<tr>
<th>Row Value</th>
<th>Column Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Row_Member 1</td>
<td>Column_Member 1 1.0</td>
</tr>
<tr>
<td>Row_Member 2</td>
<td>5.0</td>
</tr>
<tr>
<td>Row_Member 3</td>
<td>9.0</td>
</tr>
<tr>
<td>2 Row_Member 4</td>
<td>13.0</td>
</tr>
<tr>
<td>Row_Member 5</td>
<td>17.0</td>
</tr>
</tbody>
</table>

**Row Value**

**Action:**
The action is performed only when the value for the specified row satisfies the condition.
Operators:
The operators available for this function are: =, !=, <, <=, >, >=, 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:

<table>
<thead>
<tr>
<th>IF</th>
<th>Row Value</th>
<th>Value</th>
<th>10</th>
</tr>
</thead>
</table>

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.

Cross Dim Member

Action:
The action is performed only when the value in the data cell referenced by the cross-dimension member satisfies the condition. The current data cell's 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 are =, !=, <, <=, >, >=, 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:

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action:</td>
</tr>
<tr>
<td>The action is performed only 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.</td>
</tr>
</tbody>
</table>

| 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:

<table>
<thead>
<tr>
<th>IF</th>
<th>Member Name</th>
<th>Account</th>
<th>Equals</th>
<th>Value</th>
<th>Total Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Member

| Action: |
| The action is performed only 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” on page 205.

Condition Definition for Member Source Type

<table>
<thead>
<tr>
<th>Condition Definition for Member Source Type</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Condition Definition" /></td>
</tr>
</tbody>
</table>

Condition Definition for Attribute Source Type

<table>
<thead>
<tr>
<th>Condition Definition for Attribute Source Type</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Condition Definition" /></td>
</tr>
</tbody>
</table>

**Account Type**

**Action:**

The action is performed only 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:**

| Condition Definition | ![Condition Definition](image) |
**Version Type**

**Action:**
The action is performed only 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:**

<table>
<thead>
<tr>
<th>IF</th>
<th>Version Type</th>
<th>Is</th>
<th>Standard Bottom Up</th>
</tr>
</thead>
</table>

**Variance Reporting Type**

**Action:**
The action is performed only 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:**

<table>
<thead>
<tr>
<th>IF</th>
<th>Var Reporting Type</th>
<th>Is</th>
<th>Non-Expense</th>
</tr>
</thead>
</table>

**UDA**

**Action:**
The action is performed only 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:**

<table>
<thead>
<tr>
<th>IF</th>
<th>UDA</th>
<th>Scenario</th>
<th>Is</th>
<th>Value</th>
<th>ACTUAL</th>
</tr>
</thead>
</table>
Attribute

Action:
The action is performed only 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:

For additional information about data validation, see:
- “Creating and Updating Data Validation Rules” on page 211
- “Order of Evaluation and Execution for Data Validation Rules” on page 215
- “Formatting Cells and Setting the Promotional Path” on page 214
- “Viewing Data Validation Rules” on page 215
- “Conditions Supported by the Rule Builder” on page 216
- “Data Validation Conditional Operators” on page 227
- “Data Validation Rule Scenarios” on page 229

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” on page 214.

Condition Definition:

For information on other conditions, see “Conditions Supported by the Rule Builder” on page 216.

For additional information about data validation, see:
- “Creating and Updating Data Validation Rules” on page 211
- “Order of Evaluation and Execution for Data Validation Rules” on page 215
- “Formatting Cells and Setting the Promotional Path” on page 214
Range Condition Values

Subtopics

- Check Range
- Range

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” on page 226
- “Range” on page 227

For information on other conditions, see “Conditions Supported by the Rule Builder” on page 216.

For additional information about data validation, see:

- “Creating and Updating Data Validation Rules” on page 211
- “Order of Evaluation and Execution for Data Validation Rules” on page 215
- “Formatting Cells and Setting the Promotional Path” on page 214
- “Viewing Data Validation Rules” on page 215
- “Conditions Supported by the Rule Builder” on page 216
- “Data Validation Conditional Operators” on page 227
- “Data Validation Rule Scenarios” on page 229

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 \( \geq 5 \) and \(< 10 \). If this condition is met, the cell is turned red.

Condition Definition:

<table>
<thead>
<tr>
<th>Condition</th>
<th>Source Type</th>
<th>Source Value</th>
<th>Operator</th>
<th>Target Type</th>
<th>Target Value</th>
<th>Process</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHECK RANGE</td>
<td>Current Cell Value</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RANGE</td>
<td>Value</td>
<td>5</td>
<td></td>
<td>Value</td>
<td>10</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For information on other conditions, see “Conditions Supported by the Rule Builder” on page 216.

Data Validation Conditional Operators

Conditional operators in the data validation rule builder can include these types of comparisons:

- Numeric comparisons, using these operators: \( =, \neq, <, \leq, >, \geq \)
- String value comparisons, using these operators: Equals, Not Equals, Contains, Starts With, Ends With, In, and 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.

Note these considerations 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 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>
<thead>
<tr>
<th>Operator</th>
<th>Compare Value</th>
<th>Compare To Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starts With</td>
<td>2.0</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>1234.0</td>
<td>12.0</td>
</tr>
<tr>
<td></td>
<td>101.0</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>2.0</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td>2.5</td>
<td>“2.”</td>
</tr>
<tr>
<td></td>
<td>“YearTotal”</td>
<td>“Year”</td>
</tr>
<tr>
<td>Ends With</td>
<td>2.0</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td>2.0</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>2.5</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>2.5</td>
<td>“.5”</td>
</tr>
<tr>
<td></td>
<td>“YearTotal”</td>
<td>“al”</td>
</tr>
<tr>
<td></td>
<td>“YearTotal”</td>
<td>“Total”</td>
</tr>
<tr>
<td>Contains</td>
<td>2.0</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td>2.0</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>2.5</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>2.5</td>
<td>“.5”</td>
</tr>
<tr>
<td></td>
<td>2.5</td>
<td>2.5</td>
</tr>
<tr>
<td></td>
<td>23.567</td>
<td>3.5</td>
</tr>
<tr>
<td></td>
<td>23.567</td>
<td>67</td>
</tr>
<tr>
<td></td>
<td>23.567</td>
<td>“23.”</td>
</tr>
<tr>
<td></td>
<td>23.567</td>
<td>“.56”</td>
</tr>
<tr>
<td></td>
<td>“YearTotal”</td>
<td>“al”</td>
</tr>
</tbody>
</table>

For additional information about data validation, see:
- “Creating and Updating Data Validation Rules” on page 211
- “Order of Evaluation and Execution for Data Validation Rules” on page 215
Data Validation Rule Scenarios

Subtopics

- Scenario 1
- Scenario 2
- Scenario 3
- Scenario 4
- Scenario 5

These scenarios provide examples of how data validation can help implement business policies.

- “Scenario 1” on page 229
- “Scenario 2” on page 231
- “Scenario 3” on page 231
- “Scenario 4” on page 233
- “Scenario 5” on page 233

For additional information about data validation, see:

- “Creating and Updating Data Validation Rules” on page 211
- “Order of Evaluation and Execution for Data Validation Rules” on page 215
- “Formatting Cells and Setting the Promotional Path” on page 214
- “Viewing Data Validation Rules” on page 215
- “Conditions Supported by the Rule Builder” on page 216
- “Data Validation Conditional Operators” on page 227

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:

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, this 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 would be to add an IF entry to check if the Period member is YearTotal. Another option would be to split the YearTotal member into a separate column for better performance. However, this 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” on page 229.
**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 Layout at Design Time]

**Form at Data Entry Time with Data Validations Applied:**

![Form at Data Entry Time with Data Validations Applied]

For additional scenarios, see “Data Validation Rule Scenarios” on page 229.

**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:

```plaintext
@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.
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” on page 151.

- 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” on page 229.

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” on page 229.

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 ¾ 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 ¾ 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” on page 229.
# 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” on page 244). 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” on page 254.

**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” on page 277.

**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 Chapter 8, “Managing Data Validation.”

**Task Lists**

Task lists guide users through the planning process by listing tasks, instructions, and due 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.

To enable approvals notification, see “Setting Application Defaults” on page 276.

**Defining the Budgeting Process**

**Subtopics**

- Approvals Roles
- Approvals Process
- Approval Operations and Data Validations

**Approvals Roles**

To administer approvals for 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

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 the:

- 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 present on rows and columns are replaced with the distinct union of the result sets of all possible inputs. This is done with an efficient static replacement algorithm that minimizes function execution and requires no actual duplicate removal but guarantees no duplicates will be present and that all members of the distinct union will be present. User variables and context user variables on the POV will be treated as a page.
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. For **Scenario**, select a scenario.
3. For **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 planning unit title, author, 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.
5. Click **Create Report**.
   Adobe Acrobat generates a report with this planning unit information:
   - Application name
   - Selected Scenarios, Versions, and Entities
   - Planning unit title and status
   - Origin date
   - Author
   - Annotation content
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 Planning directory points to the location EPM_ORACLE_INSTANCE_HOME\common\JRE\Sun\1.6.0\lib\fonts.

### Managing Planning Unit Hierarchies

**Subtopics**
- Creating Planning Unit Hierarchies
- Assigning Planning Unit Hierarchy Scenario and Version Combinations
- Selecting Planning Unit Hierarchy Scenario and Version
- Editing Planning Unit Hierarchies
- Deleting Planning Unit Hierarchies
- Viewing Planning Unit Hierarchy Usage
- Renaming Planning Unit Hierarchies
- Synchronizing Planning Unit Hierarchies
- Exporting Planning Unit Hierarchies
- Importing 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

**Subtopics**

- **Setting Planning Unit Hierarchy Name, Scope, and Template**
- **Selecting Planning Unit Hierarchy Members**
- **Assigning Planning Unit Owners and Reviewers**
- **About Group-based Approvals**

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” on page 244).

4. Select the primary and subhierarchy members to include in the approval process (see “Selecting Planning Unit Hierarchy Members” on page 245).

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” on page 246).
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 of the planning unit hierarchy.

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.
   - Custom to define which planning units to include in the approval process.

   To add planning unit groups or individual planning units to the approval process, see “Selecting Planning Unit Hierarchy Members” on page 245.

7. In Approvals Template, select:
   - Bottom Up to use bottom-up budgeting (see “Bottom-Up Budgeting” on page 244).
   - Distribute to use distributed budgeting (see “Distributed Budgeting” on page 245).
   - Free Form to use free-form budgeting (see “Free-Form Budgeting” on page 245).

8. Take an action:
   - Click Next or select Primary and Subhierarchy Selection to select the planning unit hierarchy members (see “Selecting Planning Unit Hierarchy Members” on page 245).
   - 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” on page 243.

**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 backwards (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).
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.

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 one or more planning unit generations. Specify the generations to exclude when prompted.

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.

**Optional**: To assign planning unit reviewers and owners:

1. Take an action:

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” on page 247.

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” on page 247.

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.

4 Under Promotional Path, click to display the planning unit promotional path, verify that it is correct, and then correct any errors.

5 Under Notify These Users, click to select the users to notify for each move of the planning unit from one user to another.

6 Optional: Repeat Steps 2 through 5 for other planning units to change their inherited owners or reviewers.

7 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 1**

North America - Bill
  USA - Nick, Sandy, Kim
  CA - John

The approval path is John, Nick, Sandy, Kim, Bill.

**Example 2**

North America - Bill
  USA - (Group A)
  CA - John

The approval path is John, any user in Group A, Bill.

**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 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. From Scenario, select a scenario.
3. From 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” on page 244 to modify the planning unit hierarchy description or approvals default scope.
   - **Primary and Subhierarchy Selection**
     See “Selecting Planning Unit Hierarchy Members” on page 245 to select or remove members from approvals.
   - **Assign Owners**
     See “Assigning Planning Unit Owners and Reviewers” on page 246 to modify planning unit owners or reviewers.
   - **Usage**
     See “Viewing Planning Unit Hierarchy Usage” on page 251 to view the 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” on page 249.

### Deleting Planning Unit Hierarchies

You can delete a planning unit hierarchy if it is not referenced by other 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.

To delete planning unit hierarchies:

1. Select **Administration**, then **Approvals**, and then **Planning Unit Hierarchy** to display the planning unit hierarchy list.
2 Select the planning unit hierarchy to delete, and then click Delete.
3 Click OK to verify the deletion.

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” on page 251 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” on page 211.
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” on page 248.
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 Planning Unit Hierarchies” on page 250.

**Renaming Planning Unit Hierarchies**

You can change the name of a planning unit hierarchy.
To rename planning unit hierarchies:

1. Select **Administration**, then **Approvals**, and then **Planning Unit Hierarchy** to display the planning unit hierarchy list.
2. Select the planning unit hierarchy to rename, and then click **Rename**.
3. Enter the new name for the planning unit hierarchy in the dialog box.
4. Click **OK** to accept the new name.

**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, the new members are added as planning units only if they meet the criteria specified 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 new 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” on page 253).

See “Generating Load Files” on page 97 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. From **Existing Planning Unit Hierarchy Name**, select a planning unit hierarchy to export.
4. Click **OK**.
5. When the **Save** dialog 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.

### 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” on page 253.

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. 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 import a planning unit hierarchy:

1. Select **Administration**, then **Approvals**, and then **File Based Import/Export**.
2. Select **Import Planning Unit Hierarchy**.
3. From **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.
For Planning Unit Hierarchy with Ownership, click **Browse** to select the exported planning unit hierarchy file to import.

Click **OK**.

Click **Import** or **Done**. **Import** performs the action, and **Done** closes the dialog.

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

Subtopics

- Modifying the Planning Unit Promotional Path
- Design Considerations for Planning Unit Promotional Path Data Validation Rules

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” on page 254). 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” on page 211).
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.
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 Chapter 8, “Managing Data Validation.”

Managing Task Lists

Subtopics

- Creating Task List Folders
- Renaming Task Lists
- Renaming Task List Folders
- Moving Task List Folders
- Deleting Task List Folders
- Creating Task Lists
- Adding Instructions to Task Lists
- Adding and Defining Tasks
- Adding Tasks to Task Lists
- Editing Task Lists
- Linking Tasks to Planning Pages
- Assigning Access to Task Lists
- Importing and Exporting Task Lists

Task lists guide users through the planning process by listing tasks, instructions, and due 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.

Creating Task List Folders

To create task list folders:

1. Open the Manage Task Lists page.

   See “Managing Task Lists” on page 257.
2. In the Task List Folders area, select the folder in which to create the task list folder.
3. Click Create.
In the dialog box, enter the name of the task list, and click OK.

Renaming Task Lists

To rename task lists:
1. Open the Manage Task Lists page.
   See “Managing Task Lists” on page 257.
2. In the Task List Folders area, select the folder with the task list to rename.
3. Select the task list.
4. Select Actions, and then select Rename.
5. Enter the new task list name, then click OK.

Renaming Task List Folders

To rename task list folders:
1. Open the Manage Task Lists page.
   See “Managing Task Lists” on page 257.
2. In the Task List Folders area, select the folder to rename.
3. Select Actions, and then select Rename.
4. Enter the new Task List name, then click OK.

Moving Task List Folders

To move task list folders:
1. Open the Manage Task Lists page.
   See “Managing Task Lists” on page 257.
2. In the Task List Folders area, select the folder to move.
   You cannot move the Task Lists folder.
3. Above the Task List Folders area, click Move.
4. Select the destination folder, and click OK.

Deleting Task List Folders

To delete task list folders:
1. Open the Manage Task Lists page.
See “Managing Task Lists” on page 257.

2 In the Task List Folders area, select an empty folder to delete.
   You cannot delete the Task Lists folder.
3 Click Delete.
4 Click OK.
   If you select a folder containing additional folders, an error message is displayed.

Creating Task Lists

Task lists organize groups of tasks for users. You must create task lists before creating tasks.

To create task lists:
1 Open the Manage Task Lists page.
   See “Managing Task Lists” on page 257.
2 In the Task List Folders area, select the folder in which to create the task list.
3 Above the Task List area, select Actions, and then Create.
4 Enter the task list name, and click OK.
5 To define the task list, see:
   - “Adding Instructions to Task Lists” on page 259.
   - “Adding and Defining Tasks” on page 259.

Adding Instructions to Task Lists

To add instructions to task lists:
1 Open the Manage Task Lists page.
   See “Managing Task Lists” on page 257.
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 ” on page 260
Adding Tasks to Task Lists

You can set completion dates and alerts for tasks that users perform at runtime. Alerts display on task lists as colored circles indicated status:

- Green: On schedule
- Yellow: Approaching due date
- Red: Overdue

You can also set up email messages, for example, to alert users that a task was not completed by its due date. Alert messages are sent after an “alert date” that you set, and are repeated until the due 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” on page 257.
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 61  Task Information

<table>
<thead>
<tr>
<th>Task Type</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>URL</td>
<td>Perform these tasks:</td>
</tr>
<tr>
<td></td>
<td>● In <strong>URL</strong>, enter a fully qualified URL to associate with this task, such as <code>http://www.company_name.com</code></td>
</tr>
<tr>
<td></td>
<td>● Select <strong>Use Single Sign On</strong> to enable users to open a URL for another product that accepts single sign-on. See the <code>Oracle Enterprise Performance Management System Security Configuration Guide</code>. To link to Financial Reporting in the EPM Workspace, single sign-on is not required. Instead, include the ObjectID to link to (see the <code>Oracle Hyperion Enterprise Performance Management Workspace Administrator's Guide</code>).</td>
</tr>
<tr>
<td></td>
<td>● Go to step 7.</td>
</tr>
<tr>
<td>Form</td>
<td>Select the form for users to complete, and then go to step 7.</td>
</tr>
<tr>
<td></td>
<td><strong>Optional</strong>: Select <strong>Set Page Member Defaults</strong> 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.</td>
</tr>
<tr>
<td>Business Rule</td>
<td>Perform these tasks:</td>
</tr>
<tr>
<td></td>
<td>● From <strong>Plan Type</strong>, select the plan type associated with the business rule to execute.</td>
</tr>
<tr>
<td></td>
<td>● From <strong>Business Rules</strong>, select the business rule to execute.</td>
</tr>
<tr>
<td></td>
<td>● Go to step 7.</td>
</tr>
<tr>
<td>Manage Approvals</td>
<td>Specify the scenario and version in which the user can start the approvals process, and then go to step 7.</td>
</tr>
<tr>
<td>Job Console</td>
<td>Perform these tasks:</td>
</tr>
<tr>
<td></td>
<td>● In <strong>Job Type</strong> select the kind of job, such as to Copy Decision Package, to display in the console.</td>
</tr>
<tr>
<td></td>
<td>● In <strong>Status</strong>, select the situation of the task such as processing or in error, to display in the console.</td>
</tr>
<tr>
<td></td>
<td>● Go to step 7.</td>
</tr>
<tr>
<td>Copy Version</td>
<td>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:</td>
</tr>
<tr>
<td></td>
<td>● <strong>Scenario</strong></td>
</tr>
<tr>
<td></td>
<td>● <strong>Copy From</strong></td>
</tr>
<tr>
<td></td>
<td>● <strong>Copy To</strong></td>
</tr>
<tr>
<td></td>
<td>7 To enter a due date for the task, select <strong>Due Date</strong> and then select:</td>
</tr>
<tr>
<td></td>
<td>a. The month, day, and year. (You can change the date display format in Planning preferences. See &quot;Specifying System Settings&quot; on page 277.)</td>
</tr>
<tr>
<td></td>
<td>b. The hours, minutes, and AM or PM.</td>
</tr>
<tr>
<td></td>
<td>8 To send email message for uncompleted tasks, select <strong>Due Date</strong>, and then:</td>
</tr>
<tr>
<td></td>
<td>a. Select <strong>Repeat Every</strong>, and enter a value.</td>
</tr>
<tr>
<td></td>
<td>b. Select the frequency for email reminders.</td>
</tr>
<tr>
<td></td>
<td>9 To send email messages after the alert date and before the due date, select <strong>Alert</strong>, and then:</td>
</tr>
<tr>
<td></td>
<td>a. Set the date and time to begin sending messages by selecting the month, day, year, hours, minutes, and AM or PM.</td>
</tr>
</tbody>
</table>
b. In the Alert area, select Repeat Every and enter a value.

c. Select the frequency for email reminders.

to make task completion dependent on completing a primary task, select Dependency.

Click Instructions to enter information that explains how to complete the task.

Click Save and then OK.

Editing Task Lists

Subtopics

- Editing Tasks
- Copying Tasks
- Moving Tasks
- Moving Task Lists
- Reordering Task Lists
- Clearing Task Lists
- Deleting Tasks
- Deleting 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 due date and alert date, and email reminder messages.

To edit tasks:

1. Open the Manage Task Lists page.
   See “Managing Task Lists” on page 257.

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” on page 260.

5. For Instructions, modify instructions for completing the task.

6. Optional: To enter a due date for the task, select Due 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 due date, select Due 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 due 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 Tasks

➤ To make a copy of a task list:

1 Open the Manage Task Lists page.
   See “Managing Task Lists” on page 257.

2 Select a task list, select Actions, and then select Edit.

3 Select the task to copy, then click Save As.

4 Enter the name for the new task list, and click OK.

5 Click Close.

Moving Tasks

➤ To move tasks:

1 Open the Manage Task Lists page.
   See “Managing Task Lists” on page 257.

2 Select a task list, select Actions, and then select Edit Task List.

3 Select a task, select Actions, and then select Edit Tasks.

4 Select the task that will be a child of another task.

5 Select Actions, and then select Cut.

6 Select the task that will be the parent task.

7 Select Actions, and then Paste.

8 Click Save and Close.

➤ To cut and paste tasks:

1 Open the Manage Task Lists page.
Moving Task Lists

➢ To move task lists:
1 Open the Manage Task Lists page.
   See “Managing Task Lists” on page 257.
2 Select the folder with the task list to move.
3 Select a task list, then click Move.
4 Select the destination folder.
5 Click OK.

Reordering Task Lists

➢ To reorder task lists:
1 Open the Manage Task Lists page.
   See “Managing Task Lists” on page 257.
2 Select a task list.
3 Click ▲ or ▼.

Clearing Task Lists

After a task list is completed, you can clear the completion status, due 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 due dates. Clearing task lists disables alerts by clearing the check boxes for due dates. It does not delete the dates.

➢ To clear task lists:
1 Open the Manage Task Lists page.
   See “Managing Task Lists” on page 257.
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

● **Due Dates and Alerts**: Clears any alerts that are enabled, based on the due date set for the task

● **Both**: Clears completion status, due dates, and alerts.

4. Click the arrow to the right of the Clear drop-down list.

5. Click OK.

### Deleting Tasks

► To delete tasks:

1. Open the Manage Task Lists page.

   See “Managing Task Lists” on page 257.

2. Select the folder and task list with the task to delete, then click Edit.

3. Select the tasks to delete, select Actions, and then select Delete.

4. Click OK.

### Deleting Task Lists

► To delete task lists:

1. Open the Manage Task Lists page.

   See “Managing Task Lists” on page 257.

2. Select the folder with the task list to delete, and select the task list.

3. Select Actions, and then select Delete.

4. Click OK.

### Linking Tasks to Planning Pages

Use **Copy Link** to link tasks to 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.
Under Task - Instructions, use the browser to paste the URL link. For example, in the Microsoft Internet Explorer menu, select Edit, then Paste.

Click Save.

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” on page 257.

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 Shared Services Console to 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.
Changing Access to Task Lists

To change access to task lists:

1. Open the Manage Task Lists page.
   
   See “Managing Task Lists” on page 257.

2. Select the 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 Shared Services Console to 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. Select users or groups, and click Edit Access.

6. For Type of Access, select:
   
   - Assign: View and use
   - Manage: Modify
   - Manage and Assign: View, use, and modify
   - None: No access

7. Click Set.

8. Click Close.

Removing Access to Task Lists

To remove access to task lists:

1. Open the Manage Task Lists page.
   
   See “Managing Task Lists” on page 257.

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 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. Select the user or group, and click Remove Access.

6. Click OK.

7. Click Close.
Importing and Exporting Task Lists

Administrators can use TaskListDefUtil.cmd (Windows) or TaskListDefUtil.sh (UNIX) to move task list definitions between 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” on page 53.

To launch the TaskListDefUtil utility:

1. Enter the command from the planning1 directory, using this syntax:

```
TaskListDefUtil [-f:passwordFile] import|export FILE_NAME/TASK_LIST_NAME/-all SERVER_NAME USER_NAME APPLICATION
```

Parameter | Purpose | Required?
--- | --- | ---
[-f:passwordFile] | 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 passwordFile. See “Suppressing Password Prompts in Planning Utilities” on page 52. | No
import|export | Import or export the task list definition. | Yes
FILE_NAME/TASK_LIST_NAME/-all | When used with import, specify the XML file containing the task list definition. When used with export, specify the task list to export to XML. Use -all with import or export to import or export all XML files or task list definitions in the current application. | Yes (-all is optional)
SERVER_NAME | Server name on which the Planning application resides. | Yes
USER_NAME | Administrator’s name. | Yes
APPLICATION | When used with export, the name of the Planning application containing the task list definitions to export. When used with import, the name of the Planning application to which to import the task list definition. | Yes

2. 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:
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.
- Essbase data is copied regardless of selections for Copy Data Options.
- Because this is an administrative function, 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. **For Plan Type**, select a plan type, and 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. **For Dimension**, select a dimension from which to copy.
   b. **For 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. **For Dimensions with Source and Destination**, enter dimensions into which to copy data:
a. For **Source**, click **Member Selection**

b. For **Destination**, click **Member Selection**.

6 For **Copy Data Options**, select the type of information to copy:

- Copy Account Annotations
- Copy Supporting Details
- Copy Comments
- Copy Documents

7 Click **Copy Data**.

   Data is copied from one intersection to the other. If data is not copied successfully, a message displays. 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, Planning assumes 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” on page 290.

➤ 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. For **Dimension**, select at least one dimension with details to delete.
b. For displayed dimensions, click [open]. 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.

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:
   - Account Annotations
   - Supporting Details
   - Cell Text
   - Cell-level Document

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 displays. 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.
Managing Forms and Folders

Use the Form Management and the Business Rule Folders pages to manage folders and forms.

<table>
<thead>
<tr>
<th>Task</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create folders</td>
<td>See “Creating Folders” on page 273</td>
</tr>
<tr>
<td>Move folders</td>
<td>See “Moving Folders” on page 274</td>
</tr>
<tr>
<td>Delete folders</td>
<td>See “Deleting Folders” on page 274</td>
</tr>
<tr>
<td>Create forms</td>
<td>See “Creating Simple Forms” on page 159</td>
</tr>
<tr>
<td>Assign access to forms and folders</td>
<td>See “Assigning Access to Forms and Folders” on page 62</td>
</tr>
<tr>
<td>Move forms</td>
<td>See “Moving Forms and Ad Hoc Grids” on page 157</td>
</tr>
<tr>
<td>Delete forms</td>
<td>See “Deleting Forms” on page 157</td>
</tr>
<tr>
<td>Rename folders</td>
<td>See “Renaming Folders” on page 275</td>
</tr>
</tbody>
</table>

To view all forms or business rules in a 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 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**.

### Moving Folders

When you move folders, all nested folders, forms, and Calculation Manager business rules within them are also moved.

To move 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.
3. Click **Move**.
4. Select the destination folder to which to move the selected folder.
5. Click **OK**.

### Deleting Folders

To delete folders:

1. For form folders: Select **Administration**, then **Manage**, then **Forms and Ad Hoc Grids**.
   - For Calculation Manager business rule folders: Select **Administration**, then **Business Rule Security**.
2. Select the folder to delete.
3. Click **Delete**.
4. Click **OK**.
Renaming Folders

To rename folders:

1. For form folders: Select Administration, then Manage, and then Forms and Ad Hoc Grids.
   For Oracle Hyperion Calculation Manager business rule folders: Select Administration, then Business Rule Security.
2. Select the folder to rename.
3. Click Rename.
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” on page 276
- “Setting Application Defaults” on page 276
- “Specifying System Settings” on page 277
- “Limiting Use of an Application During Maintenance” on page 278
- “Specifying Custom Tools” on page 280
- “Setting Display Options” on page 280
- “Setting Printing Options” on page 281

Administrators can control which tabs display for setting personal preferences, application defaults, and system settings.

<table>
<thead>
<tr>
<th>Show Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current Application Defaults</td>
<td>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 Oracle Hyperion Planning User's Guide.</td>
</tr>
</tbody>
</table>
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 Preferences in the left hand pane, or select File, then Preferences.
2. Click Planning, 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” on page 192.

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 outlined below, 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 such as:
     - Email notifications for task lists, approvals, and job console
     - 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 such as
Specifying System Settings

Only administrators can specify system-wide settings.

Only the application owner can set the Email Server and Password for Synchronizing With Essbase.

To specify system settings:

1. Select Administration, then Application, then Settings.
2. Select Advanced Settings.
4. Set options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Email Character Set</strong></td>
<td>The character set for email messages:</td>
</tr>
<tr>
<td></td>
<td><strong>UTF-8</strong>: Unicode encoding format</td>
</tr>
<tr>
<td></td>
<td><strong>Regional Setting</strong>: The system’s regional setting</td>
</tr>
<tr>
<td><strong>Shared Services URL</strong></td>
<td>The URL for the Shared Services server. Click Register Shared Services and assign the application to an application group (see “Assigning Applications to Shared Services Application Groups” on page 278).</td>
</tr>
<tr>
<td><strong>Enable Use of the Application for</strong></td>
<td>Determine whether users can access the application in maintenance mode, such as during backups. See “Limiting Use of an Application During Maintenance” on page 278.</td>
</tr>
<tr>
<td><strong>Enable Display of Substitution Variables</strong></td>
<td>Set how substitution variables display in the Member Selection dialog box when users respond to runtime prompts in business rules:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Display All</strong>: Display all substitution variables</td>
</tr>
<tr>
<td></td>
<td>• <strong>Display None</strong>: Do not display substitution variables</td>
</tr>
<tr>
<td></td>
<td>• <strong>Enable Filtering</strong>: Display only substitution variables that are valid for the runtime prompt</td>
</tr>
<tr>
<td><strong>Display Users’ Full Names</strong></td>
<td>Yes: Show full names (such as Victoria Hennings).</td>
</tr>
<tr>
<td></td>
<td><strong>No</strong>: Do not display full names. Only user IDs will be shown (such as VHennings).</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Calculation Module</td>
<td>The module used to create, validate, deploy, and administer business rules. (Calculation Manager is the only option.)</td>
</tr>
<tr>
<td>Select User</td>
<td>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.)</td>
</tr>
</tbody>
</table>

5 To specify system settings for each application, repeat these steps, then click **Save**.

### Assigning Applications to Shared Services Application Groups

To assign a Planning application to a 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**, 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” on page 279.

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” on page 53.

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 63

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Required?</th>
</tr>
</thead>
<tbody>
<tr>
<td>[-f:passwordFile]</td>
<td>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 passwordFile. See “Suppressing Password Prompts in Planning Utilities” on page 52.</td>
<td>No</td>
</tr>
<tr>
<td>/A=app</td>
<td>Application name</td>
<td>Yes</td>
</tr>
<tr>
<td>/U=user</td>
<td>Name of the administrator executing the utility</td>
<td>Yes</td>
</tr>
<tr>
<td>/P=password</td>
<td>The administrator’s password</td>
<td>Yes</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td>Required?</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>/LL=loginLevel</td>
<td>Specify which users the utility affects:</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>ALL_USERS - All users can log on or continue working with the application.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>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.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>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.</td>
<td></td>
</tr>
<tr>
<td>/DEBUG=[true</td>
<td>false]</td>
<td>Specify whether to run the utility in debug mode. The default is false.</td>
</tr>
<tr>
<td>/HELP=Y</td>
<td>View the utility syntax online.</td>
<td>No</td>
</tr>
</tbody>
</table>

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**, 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 Other Options” on page 167.

Mapping an Application for Reporting

Subtopics
- 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
- Validation Rules for Default Members
- Pushing Data to a Reporting Application
- Pushing Data with a Utility
- 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 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, 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 Chapter 5, “Loading Data and Metadata.”

Next:

- Map the source Planning application to the target reporting application.
  
  See “Mapping a Planning Application to a Reporting Application” on page 282.

- Push data from the source to the target application.
  
  See “Pushing Data to a Reporting Application” on page 285.

- Report on the data in the reporting application.

**Mapping a Planning Application to a Reporting Application**

> To map a Planning application to a reporting application:

1. **Create the reporting application.**
   
   See “Creating a Reporting Application” on page 282.

2. **In Planning, select Administration, then Map Reporting Applications.**

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” on page 283.
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” on page 285.

To update the mapping for Smart Lists, see “Synchronizing Smart Lists in Reporting Applications” on page 364.

### Defining Application Mappings

To define an application mapping:

1. **For new mappings**: Type 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 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” on page 289, “Editing an Essbase Server for Reporting Applications” on page 289, or “Removing an Essbase Server for Reporting Applications” on page 290.
5. Click **Next**.

See “Defining Dimension Mappings” on page 283.

### Defining Dimension Mappings

On the **Map Dimensions** tab, you map the 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” on page 284).

**Note**: If parent members were enabled for dynamic members, and the resulting placeholders for dynamic member exist in the aggregate storage cube, Planning member names need not exist in aggregate storage cube. However, if only Planning member names exists in aggregate storage cube, the Push Data operation will fail because the system requires the placeholder names generated for dynamic members.

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. **For 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” on page 364.

3. **For 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” on page 284.
   - **Save** if all dimensions are mapped and there is no need for a POV. (If so, the **Next** button is grayed out.)

   You can click **Save As** to save the application mapping with a new name, and then click **Save** in the **Save As** dialog box.

### 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 Planning dimension. Either:
   - Type 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” on page 285.

2 In the bottom section, select a member for each unmapped reporting application POV member. Either:
   - Type the POV member name.
   - To select members that are not displayed, click ![select icon], and then select the POV member.

3 Click Save.

   You can click Save As to save the application mapping with a new name, and then click Save in the Save As dialog box.

   The members are checked for validity. See “Validation Rules for Default Members” on page 285.

Validation Rules for Default Members

Default members in the reporting application store the data that is pushed from the source 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.

Pushing Data to a Reporting Application

After setting up application mappings, you can push data to a reporting application. Planning validates the selected application mappings, and then pushes the mapped Planning dimension data to the reporting application dimensions. A progress bar indicates the transfer status. You
can also check the Job Console for the job status. See “Mapping a Planning Application to a Reporting Application” on page 282.

Note: If you enabled parent members for dynamic on-the-fly member creation, and placeholders consequently exist for dynamic members in Essbase, the Push Data operation fails if only Planning member name exists in the aggregate storage application.

If only planning member name exists in aggregate storage application, Push Data fails because it is looking for the bucket name.

To push data to a reporting application:

1. Create the reporting application.
   See “Creating a Reporting Application” on page 282.

2. In Planning, select Administration, and then Map Reporting Applications.


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.
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” on page 53.

To launch the **PushData** utility:

1 **From the `planning1` directory on the server where Planning is installed, enter this syntax at the Command Prompt:**

   ```bash
   ```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f:passwordFile</code></td>
<td><strong>Optional</strong>: 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” on page 52.</td>
</tr>
<tr>
<td><code>/U:username</code></td>
<td>The name of the Planning administrator</td>
</tr>
<tr>
<td><code>/A:sourceApplication</code></td>
<td>The name of the source Planning application from which to push data</td>
</tr>
<tr>
<td><code>/M:applicationMapping</code></td>
<td>The name of the application mapping that defines the push data range (see “Mapping an Application for Reporting” on page 281)</td>
</tr>
<tr>
<td><code>[/C]</code></td>
<td><strong>Optional</strong>: Clears the data range in the target reporting application before pushing data</td>
</tr>
</tbody>
</table>

2 **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” on page 53.
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 Essbase substitution variables defined only for the 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 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:

   ```
   ```

   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 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 Planning.

**Tip:** Using the Planning Upgrade Wizard, you can update Essbase server information for multiple reporting applications (see “Updating References to Reporting Essbase Servers” on page 308).

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:

```
```

Entering the Essbase cluster name without the APS URL in the **Essbase Server name** field is not supported in this release.

1. **In Server Name**, type the server name.
2. **In User Name**, type your user name.
3. **In Password**, type your password.
4. **Optional**: To test the connection to the Essbase server, click **Validate Connection**.
5. Click **OK**.

---

**Removing an Essbase Server for Reporting Applications**

To remove an 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 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**

Planning provides SQL files to delete this information:

- Account annotations. See “Deleting Account Annotations” on page 290.
- Supporting detail associated with scenarios. See “Deleting Supporting Detail Associated With a Scenario” on page 291.

You can use the Clear Cell Detail feature to clear account annotations, supporting detail, cell text, and cell-level documents. See “Clearing Cell Details” on page 270.

---

**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” on page 270.

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`:

```sql
DELETE
FROM HSP_ACCOUNT_DESC
WHERE ACCOUNT_ID=(SELECT OBJECT_ID FROM HSP_OBJECT
WHERE OBJECT_NAME='ACCOUNT1')
```

```sql
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 `sddelete.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” on page 270.

To delete supporting detail associated with scenarios:

1. Stop the Web application server.
2. Update the SQL queries section of the `sddelete.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 `sddelete.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:

```sql
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');
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 click OK.
   - To change a menu, select it and click Edit.
   - To delete menus, select them, click Delete, and 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, 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 AddSibling.
   - 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, 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>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Menu Item</td>
<td>Enter a unique name containing only alphanumeric and underscore characters, with no special characters or spaces</td>
</tr>
<tr>
<td>Label</td>
<td>Enter text to display 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 File directly or to the name of a resource, such as LABEL_FILE, which can be localized.</td>
</tr>
</tbody>
</table>
**Icon**

Optional: In context of the 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: Images\green.gif

**Type**

Select the menu item type to determine available Properties. No properties are available for Menu Header.

- **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**

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.

**Define menu item properties, which differ for menu item types:**

<table>
<thead>
<tr>
<th>Type</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Form</strong></td>
<td>a. From Form Folder, select the folder containing the destination form.</td>
</tr>
<tr>
<td></td>
<td>b. From Form, select the form.</td>
</tr>
<tr>
<td><strong>URL</strong></td>
<td>a. For URL, enter the complete URL to which to direct the user. For example: <a href="http://server">http://server</a> name/HFM/Logon/HsvLogon.asp.</td>
</tr>
<tr>
<td></td>
<td>b. Select Use Single Sign-on to append the SSO token to the URL.</td>
</tr>
<tr>
<td></td>
<td>c. Select Include Context in URL to include the context.</td>
</tr>
<tr>
<td><strong>Business Rule</strong></td>
<td>a. For Plan Type, select the plan type for which the business rule is available.</td>
</tr>
<tr>
<td></td>
<td>b. For Business Rules, select the business rule to launch.</td>
</tr>
<tr>
<td></td>
<td>c. From View Type, select how to display runtime prompt pages:</td>
</tr>
<tr>
<td></td>
<td>● Classic View: Use the default Planning view</td>
</tr>
<tr>
<td></td>
<td>● Streamline View: Display each runtime prompt on a different line</td>
</tr>
<tr>
<td></td>
<td>d. Optional: For Window Title, enter a title to display instead of Runtime Prompts.</td>
</tr>
<tr>
<td></td>
<td>e. Optional: For OK Button Label, enter the text to display for the OK button.</td>
</tr>
<tr>
<td></td>
<td>f. Optional: For Cancel Button Label, enter the text to display for the Cancel button.</td>
</tr>
<tr>
<td></td>
<td>g. Optional: For 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.</td>
</tr>
<tr>
<td>Type</td>
<td>Options</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Manage Approvals</td>
<td>Specify the planning unit to which the user is directed by selecting a scenario and a version.</td>
</tr>
<tr>
<td>Previous Form</td>
<td>Enter the name of the menu item that will return the user to the previous form.</td>
</tr>
</tbody>
</table>
| Copy Version     | 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:  
|                  | a. From **Scenario**, select the scenario from which to copy.  
|                  | b. From **Copy From**, select the version that contains the data to copy.  
|                  | c. From **Copy To**, select the version to which to copy the data. |
| Job Console      | Enable end users to view the Job Console for the specified job type and job. Select the following default values:  
|                  | a. From **Job Type**, select one:  
|                  |   ● **Select Job Type**  
|                  |   ● **Business Rule**  
|                  |   ● **Ruleset**  
|                  |   ● **Sequence**  
|                  |   ● **Clear Cell Details**  
|                  |   ● **Copy Data**  
|                  |   ● **Push Data**  
|                  | b. From **Status**, select one:  
|                  |   ● **Select Run Status**  
|                  |   ● **Status Processing**  
|                  |   ● **Status Completed**  
|                  |   ● **Status Error** |

6 Click Save.
About Creating Applications with Performance Management Architect and Planning Application Administration

You can create applications using Performance Management Architect or Planning application administration. Different menus and options are available for each type of application. For example, for applications created with Performance Management Architect application administration, you manage Smart Lists within Performance Management Architect. For applications created with Planning application administration, you can select Administration, then Manage, then Smart Lists to use the feature within Planning. Planning application administration tasks are described in this chapter. 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.

For information on setting up applications for Planning modules, see their respective guides.

Setting Up Applications Using Planning Application Administration

If you are using Planning application administration, use the Planning Application Wizard to create and delete applications, and register with Shared Services.
To set up applications using Planning application administration:

1. **Perform one task:**
   - From Planning, select **Administration**, then **Application**, and then **Manage Applications**.
   - From EPM Workspace, select **Navigate**, then **Administer**, and then **Planning Administration**.

2. **See these topics to complete the task:**
   - “Managing Applications” on page 298
   - “Managing Data Sources” on page 303
   - “Managing Upgrades” on page 306

### Managing Applications

To use Planning application administration to create, delete, and register Planning applications, see:

- “Creating Applications” on page 298
- “Deleting Applications” on page 302
- “Registering Applications” on page 303

### Creating Applications

To create and update applications using Planning application administration:

1. **Start the Planning Application Wizard** (see “Setting Up Applications Using Planning Application Administration” on page 297).

2. **In Manage Applications**, click **Actions**, and then select **Create**.

3. **Define the application by completing information in the tabs.** See:
   - “Selecting Application Information” on page 298
   - “Setting up the Calendar” on page 299
   - “Setting up Currencies” on page 301
   - “Specifying Plan Types” on page 301
   - “Reviewing Application Information” on page 302

### Selecting Application Information

Specify the application name and description, register the application with Shared Services and select the data source.

A default instance (cluster) is set up when you install and configure 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” on page 297.

2. Click Select.

3. Select a data source.

   See “Managing Data Sources” on page 303.

4. Enter the name of the application.

   The name can contain up to eight characters. It must not be the same as an existing 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. 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” on page 369.

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>
<thead>
<tr>
<th>Fiscal Year First Month</th>
<th>Fiscal Start Year</th>
<th>Period - Year</th>
<th>Years Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>January</td>
<td>Same Calendar Year</td>
<td>Jan-12 to Dec-12</td>
<td>FY12</td>
</tr>
<tr>
<td>July</td>
<td>Same Calendar Year</td>
<td>Jul-12 to Jun-13</td>
<td>FY12</td>
</tr>
<tr>
<td>Fiscal Year First Month</td>
<td>Fiscal Start Year</td>
<td>Period - Year</td>
<td>Years Dimension</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------------</td>
<td>---------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>July</td>
<td>Previous Calendar Year</td>
<td>Jul-11 to Jun-12</td>
<td>FY12</td>
</tr>
<tr>
<td>February</td>
<td>Same Calendar Year</td>
<td>Feb-12 to Jan-13</td>
<td>FY12</td>
</tr>
<tr>
<td>February</td>
<td>Previous Calendar Year</td>
<td>Feb-11 to Jan-12</td>
<td>FY12</td>
</tr>
<tr>
<td>December</td>
<td>Same Calendar Year</td>
<td>Dec-12 to Nov-13</td>
<td>FY12</td>
</tr>
<tr>
<td>December</td>
<td>Previous Calendar Year</td>
<td>Dec-11 to Nov-12</td>
<td>FY12</td>
</tr>
</tbody>
</table>

To set up the calendar:

1. In the Planning Application Wizard, click Calendar.
   See “Setting Up Applications Using Planning Application Administration” on page 297.

2. 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.

3. 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.

4. 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.

5. 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.

6. 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 2010 with a starting period of Jun creates the starting year as FY10, defined as July-10 to Jun-11.
   - **Previous Calendar Year**: Set the year to start with the previous calendar year. For example, selecting the year 2012 with a starting period of Jun creates the starting year as FY12, defined as Jul-11 to Jun-12.

   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**.

---

300 Working with Planning Application Administration
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.

Note: All Years is available only for applications created for Planning modules, not for generic Planning applications.

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 Planning Application Wizard, click Currencies.

See “Setting Up Applications Using Planning Application Administration” on page 297.

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 Essbase database is created for each plan type. You cannot change the name or number of plan types 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. Public Sector Planning and Budgeting and Oracle Project Financial Planning users will need to use Oracle Hyperion Workforce Planning and/or Oracle Hyperion Capital Asset Planning plan types and should ensure those plan types are tagged as such so they will not count towards the three generic total. 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” on page 297.

2. **For each plan type in the application, select the Plan Type and specify a plan name.**
   
   You must select at 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 Planning Application Wizard, click Finish.**
   
   See “Setting Up Applications Using Planning Application Administration” on page 297.

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 Planning Application Wizard to delete applications that use Planning application administration.

| Caution! | Oracle recommends backing up the application before deleting. See “Back Up Applications and Application Databases” on page 91. |
To delete applications:

1. Back up the application.
2. Start the Planning Application Wizard (see “Setting Up Applications Using Planning Application Administration” on page 297).
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 Planning Application Wizard to reregister the Shared Services application group for an application.

To register applications:

1. Start the Planning Application Wizard (see “Setting Up Applications Using Planning Application Administration” on page 297).
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 Planning application must be associated with a data source, which links the relational database and the Essbase server. To use the Planning Application Wizard to create, update, test, and delete data sources, see:

- “Creating Data Sources” on page 303
- “Editing Data Sources” on page 305
- “Checking Connections” on page 306
- “Deleting Data Sources” on page 306


Data sources must be associated with instances, also called clusters. To update clusters, see the Oracle Hyperion Enterprise Performance Management Architect Administrator’s Guide.

Note: If you use the EPM System Configurator to change the Essbase server, update the server name within Planning.

Creating Data Sources

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 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” on page 297. 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 Planning application administration, you can use the wizard to update the data source name, description, relational database, and details for the relational database and Essbase server.

To edit data sources:

1. In the Planning Application Wizard, click **Manage Data Source**. See “Setting Up Applications Using Planning Application Administration” on page 297. 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.

12 Use the Oracle Hyperion Enterprise Performance Management System Configurator to select an instance (cluster) to use for the application. See the Oracle Hyperion Enterprise Performance Management Architect Administrator’s Guide.

Checking Connections

You can test the connections to the database or Essbase.

To check connections:

1 In the Planning Application Wizard, click Manage Data Source.
   See “Setting Up Applications Using Planning Application Administration” on page 297.
2 To test the connection, click Validate.
3 At the message that the connection is successful, click the X to close the message.

Deleting Data Sources

For 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” on page 297. 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 Planning Upgrade Wizard to update references to rehosted data sources and to upgrade Planning applications. For example, if the 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 created in Performance Management Architect or applications using Planning application administration.
For updating references to rehosted relational databases and Essbase servers, see “Updating References to Data Sources” on page 307.

For updating references to rehosted Essbase servers that have reporting applications, see “Updating References to Reporting Essbase Servers” on page 308.

For upgrading Planning applications created in Performance Management Architect or Planning application administration, see “Upgrading Applications” on page 308.

**Updating References to Data Sources**

Use the Planning Upgrade Wizard to update references to rehosted relational databases and 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:


     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” on page 308).

**Note:** If you do not see the updates reflected in the Planning application, stop and then restart the Planning server.

**Upgrading Applications**

Use the Planning Upgrade Wizard to upgrade Planning applications created using Performance Management Architect or 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” on page 307 and “Updathing References to Reporting Essbase Servers” on page 308.

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 Planning Upgrade Wizard to update references to 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” on page 308).

**Note:** If you do not see the updates reflected in the Planning application, stop and then restart the Planning server.

**Working with Alias Tables**

You can create and update alias tables, and set a default alias table for the application. Follow naming conventions in Appendix B, “Naming Restrictions.”

**About Alias Tables**

You can assign alternate names, or aliases, to 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, 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 these language combinations:

- English, French, German, Spanish, and Italian
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
Working with Dimensions

Subtopics

- Dimension Overview
- Working with Dimension Hierarchies
- Viewing a Member’s Ancestors
- Determining Where Members Are Used in an Application
- About Custom Dimensions
- About Entities
- Base Currency
- About Accounts
- Accounts, Entities, and Plan Types
- About User-Defined Custom Dimensions
- Adding or Editing User-Defined Custom Dimensions
- Working with Members
- Working with Attributes
- Working with Attribute Values
- Customizing Calendars
- Setting Up Currencies
- Enabling Multiple Currencies
- Specifying Exchange Rates
- Setting Up Scenarios
- Specifying Versions
- Sorting Version and Scenario Members
- Moving Scenario and Version Members in the Dimension Hierarchy

Dimension Overview

Dimensions categorize data values. Seven dimensions are included with 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. 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” on page 90 and “Setting Dimension Density and Order” on page 325.

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify or change dimension properties.</td>
<td>Click Edit</td>
</tr>
<tr>
<td>Add a dimension.</td>
<td>Click Add Dimension</td>
</tr>
<tr>
<td>Search for a dimension member.</td>
<td>See “Finding Dimensions or Members” on page 315</td>
</tr>
<tr>
<td>Expand or collapse the dimension hierarchy.</td>
<td>Click Expand or Collapse</td>
</tr>
<tr>
<td>Add or edit a dimension member.</td>
<td>Click Add Child or Add Sibling</td>
</tr>
<tr>
<td>Move a dimension member.</td>
<td>See “Moving Members Within the Dimension Hierarchy” on page 315</td>
</tr>
<tr>
<td>Delete a dimension member.</td>
<td>See “Deleting Members” on page 329</td>
</tr>
<tr>
<td>Assign access to a dimension member.</td>
<td>See “Assigning Access to Members and Business Rules” on page 58</td>
</tr>
<tr>
<td>View a member’s ancestors.</td>
<td>Click Show Ancestors</td>
</tr>
</tbody>
</table>
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.

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 [v] 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 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

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>
<thead>
<tr>
<th>Option</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic Calc and Store</td>
<td>Calculates data values of members, and stores values.</td>
</tr>
<tr>
<td>Store</td>
<td>Stores data values of members.</td>
</tr>
<tr>
<td>Dynamic Calc</td>
<td>Calculates data values of members, and disregards the values.</td>
</tr>
<tr>
<td>Never Share</td>
<td>Prohibits members in the same dimension from sharing data values.</td>
</tr>
<tr>
<td>Shared</td>
<td>Allows members in the same dimension to share data values.</td>
</tr>
<tr>
<td>Label Only</td>
<td>Has no data associated with the member.</td>
</tr>
</tbody>
</table>

About Dynamic Calc

With dynamically calculated members, 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” on page 237).

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, 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 67  Using Account Types

<table>
<thead>
<tr>
<th>Account Type</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expense</td>
<td>Cost of doing business</td>
</tr>
<tr>
<td>Revenue</td>
<td>Source of income</td>
</tr>
<tr>
<td>Asset</td>
<td>Company resource</td>
</tr>
<tr>
<td>Liability and Equity</td>
<td>Residual interest or obligation to creditors</td>
</tr>
<tr>
<td>Saved assumption</td>
<td>Centralized planning assumptions ensuring consistency across the application</td>
</tr>
</tbody>
</table>

Summary of Account Types

Table 68  Summary of Account Types

<table>
<thead>
<tr>
<th>Account Type</th>
<th>Time Balance</th>
<th>Variance Reporting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Revenue</td>
<td>Flow</td>
<td>Non-Expense</td>
</tr>
<tr>
<td>Expense</td>
<td>Flow</td>
<td>Expense</td>
</tr>
<tr>
<td>Asset</td>
<td>Balance</td>
<td>Non-Expense</td>
</tr>
<tr>
<td>Liability</td>
<td>Balance</td>
<td>Non-Expense</td>
</tr>
<tr>
<td>Equity</td>
<td>Balance</td>
<td>Non-Expense</td>
</tr>
<tr>
<td>Saved Assumption</td>
<td>User-defined</td>
<td>User-defined</td>
</tr>
</tbody>
</table>

Variance reporting and time balance settings are system-defined; only Saved Assumption is user-defined.

Time Balance Property

Time balance specifies how Planning calculates the value of summary time periods.

Table 69  Time Balance Properties

<table>
<thead>
<tr>
<th>Time Balance Property</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow</td>
<td>Aggregate of all values for a summary time period as a period total.</td>
<td>Jan: 10 Feb: 15 Mar: 20 Q1: 45</td>
</tr>
<tr>
<td>First</td>
<td>Beginning value in a summary time period as the period total.</td>
<td>Jan: 10 Feb: 15 Mar: 20 Q1: 10</td>
</tr>
<tr>
<td>Balance</td>
<td>Ending value in a summary time period as the period total.</td>
<td>Jan: 10 Feb: 15 Mar: 20 Q1: 20</td>
</tr>
<tr>
<td><strong>Time Balance Property</strong></td>
<td><strong>Description</strong></td>
<td><strong>Example</strong></td>
</tr>
<tr>
<td>---------------------------</td>
<td>-----------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Average</td>
<td>Average for all the child values in a summary time period as the period total.</td>
<td>Jan: 10 Feb: 15 Mar: 20 Q1: 15</td>
</tr>
<tr>
<td>Fill</td>
<td>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.</td>
<td>Jan: 10; Feb: 10; Mar: 10; Q1: 30</td>
</tr>
<tr>
<td>Weighted Average - Actual_Actual</td>
<td>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 \times 31 + 15 \times 29 + 20 \times 31) / 91 = 15)</td>
<td>Jan: 10 Feb: 15 Mar: 20 Q1: 15</td>
</tr>
<tr>
<td>Weighted Average - Actual_365</td>
<td>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 \times 31 + 15 \times 28 + 20 \times 31) / 90 = 15)</td>
<td>Jan: 10 Feb: 15 Mar: 20 Q1: 15</td>
</tr>
</tbody>
</table>

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>
<thead>
<tr>
<th><strong>Table 70  Effect of Skip Options When Time Balance is Set to First</strong></th>
<th><strong>Example</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Skip Option</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td><strong>None</strong></td>
<td>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.</td>
</tr>
<tr>
<td>Skip Option</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Missing</td>
<td>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 the calculating parent values, so Q1 = second child (Feb), or 20.</td>
</tr>
<tr>
<td>Zeros</td>
<td>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.</td>
</tr>
<tr>
<td>Missing and Zeros</td>
<td>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.</td>
</tr>
</tbody>
</table>

### 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, 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 Appendix B, “Naming Restrictions.”

Table 71  Properties for User-Defined Custom Dimensions

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimension</td>
<td>Enter a name that is unique across all dimensions.</td>
</tr>
<tr>
<td>Alias</td>
<td><strong>Optional:</strong> Select an alias table. Enter a unique alternate name for the dimension.</td>
</tr>
<tr>
<td>Description</td>
<td><strong>Optional:</strong> Enter a description.</td>
</tr>
<tr>
<td>Valid for Plan Types</td>
<td>Select plan types for which the dimension is valid. Clearing this option makes all members of the dimension invalid for the deselected plan type.</td>
</tr>
<tr>
<td>Apply Security</td>
<td>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.</td>
</tr>
<tr>
<td>Data Storage</td>
<td>Select a data storage option. The default is Never Share.</td>
</tr>
</tbody>
</table>

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 Appendix B, “Naming Restrictions.”

Table 72  Dimension Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimension</td>
<td>Enter a dimension name.</td>
</tr>
<tr>
<td>Alias</td>
<td><strong>Optional:</strong> Select an alias table and enter an alternate name of up to 80 characters. Follow the dimension naming restrictions.</td>
</tr>
<tr>
<td>Description</td>
<td><strong>Optional:</strong> Enter a description.</td>
</tr>
<tr>
<td>Valid for Plan Types</td>
<td>Select plan types for which the dimension is valid. Not available for Entity or Account dimensions.</td>
</tr>
<tr>
<td>Property</td>
<td>Value</td>
</tr>
<tr>
<td>--------------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>Apply Security</td>
<td>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.</td>
</tr>
<tr>
<td>Data Storage</td>
<td>Select data storage options.</td>
</tr>
<tr>
<td>Display Option</td>
<td>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.</td>
</tr>
<tr>
<td>Enable custom attribute display</td>
<td>Display available and selected attributes for dimensions with associated attributes. Enable custom attribute display for dimensions with attributes.</td>
</tr>
</tbody>
</table>

**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” on page 312.
  4. Set the order of precedence by selecting a dimension and clicking or .
     - See “About Reordering Dimensions” on page 90.

**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

Subtopics
- About Dynamic Members
- Adding or Editing Members
- Deleting Members
- Deleting Parent Members
- Viewing Member Properties from Forms
- Working with Shared Members
- Creating Shared 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 those that users can create when working with business rules if prompted that insufficient members exist, and that an administrator must refresh the database to recreate the required placeholders in Essbase. For information about working with business rules and dynamic members, see the Oracle Hyperion Calculation Manager Administrator Features.

Note: The ability to create dynamic members is only available in initialized module applications such as Workforce Planning, Public Sector Planning and Budgeting, Capital Asset Planning, and Project Financial Planning.

To configure dimensional member parents to support dynamic members:

- Edit the member parents as described in “Adding or Editing Members” on page 327.
- Specify the last three properties in Table 73.
- Refresh the database to create the placeholders for dynamic members in Essbase.

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 Essbase. Once 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.

### Adding or Editing Members

Members must conform to guidelines listed in Appendix B, “Naming Restrictions.” Shared members must be consistent with “Working with Shared Members” on page 331.

#### Table 73 Member Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
<td>Enter a name that is unique across all dimension members.</td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>Optional: Enter a description.</td>
</tr>
<tr>
<td><strong>Alias Table</strong></td>
<td>Optional: Select the alias table to store the alias name. Enter an alternate name for the member.</td>
</tr>
<tr>
<td><strong>For Account members only: Account Type</strong></td>
<td>Select the account type.</td>
</tr>
<tr>
<td><strong>For Account members only: Variance Reporting</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>For Account members only: Time Balance</strong></td>
<td>For Time Balance, select Flow or Balance.</td>
</tr>
<tr>
<td><strong>For Account members only: Skip</strong></td>
<td>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” on page 321.</td>
</tr>
<tr>
<td><strong>For Account members only: Exchange Rate Type</strong></td>
<td>For Exchange Rate Type, select an option.</td>
</tr>
<tr>
<td><strong>For Account members only: Data Type</strong></td>
<td>For Data Type, select a data type.</td>
</tr>
<tr>
<td><strong>Distribution</strong></td>
<td>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.</td>
</tr>
<tr>
<td>Property</td>
<td>Value</td>
</tr>
<tr>
<td>----------</td>
<td>-------</td>
</tr>
<tr>
<td><strong>Hierarchy Type</strong></td>
<td>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. <strong>Note:</strong> 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.</td>
</tr>
<tr>
<td><strong>Data Storage</strong></td>
<td>Select a data storage property. The default is Never Share for new custom dimension members (except root members).</td>
</tr>
<tr>
<td><strong>Two Pass Calculation</strong></td>
<td>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.</td>
</tr>
<tr>
<td>For Entity members only: Base Currency</td>
<td>Select the Entity member’s base currency.</td>
</tr>
<tr>
<td><strong>Display Option</strong></td>
<td>Set application default display options for the Member Selection dialog box. Select <strong>Member Name</strong> or <strong>Alias</strong> to display members or aliases. <strong>Member Name:Alias</strong> displays members on the left and aliases on the right. <strong>Alias:Member Name</strong> displays aliases on the left and members on the right.</td>
</tr>
<tr>
<td><strong>Plan Type</strong></td>
<td>Select plan types for which the member is valid. <strong>Note:</strong> 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. <strong>Caution!</strong> 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.</td>
</tr>
<tr>
<td>For Entity members only: Base Currency</td>
<td>Select the base currency for the Entity member.</td>
</tr>
<tr>
<td>For Account members only: Source Plan Type</td>
<td>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.</td>
</tr>
<tr>
<td><strong>Smart Lists</strong></td>
<td><strong>Optional:</strong> Select a Smart List to associate with the member.</td>
</tr>
<tr>
<td><strong>Enable for Dynamic Children</strong></td>
<td>Allows users to create additional children for this member using business rules.</td>
</tr>
<tr>
<td><strong>Number of Possible Dynamic Children</strong></td>
<td>Enter the maximum number of members that users can create as needed or as prompted.</td>
</tr>
<tr>
<td>Property</td>
<td>Value</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| **Access Granted to Member Creator** | Determines the access that users will have if they dynamically create members when launching business rules:  
  - **Inherit**—They will be able to access the closest member in hierarchy.  
  - **None**—They cannot view members  
  - **Read**—They can view but not modify new, dynamically created members  
  - **Write**—They can modify new, dynamically created, 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:
   - 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.
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 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” on page 237.

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.

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 properties 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 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 Smart View.

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” on page 333. Attribute names must conform to guidelines listed in Appendix B, “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” on page 333. 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 ".<n> 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 Planning database. Use the Years dimension to add years to the calendar.

**Table 74  Calendar Tasks**

<table>
<thead>
<tr>
<th>Task</th>
<th>See Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Work with the fiscal year and period hierarchy.</td>
<td>“Working with the Years Dimension” on page 338.</td>
</tr>
</tbody>
</table>

**Defining How Calendars Roll Up**

**Table 75  Calendar Roll Up**

<table>
<thead>
<tr>
<th>Base Time Period</th>
<th>Roll Up</th>
</tr>
</thead>
<tbody>
<tr>
<td>12 Months</td>
<td>Four quarters are created per year. Months roll up into parent quarters and quarters roll up into years.</td>
</tr>
<tr>
<td>Quarters</td>
<td>Quarters roll up into years.</td>
</tr>
<tr>
<td>Custom</td>
<td>No default rollup structures. A flat list of the custom base time periods displays.</td>
</tr>
</tbody>
</table>

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.

You can have up to 100 calendar years and 500 time periods in an application. Actual limits are a function of calendar years and time periods. How many time periods and years you can set also depends on whether your application uses multiple currencies. Oracle recommends these practical limits for an application:

- 400 time periods per year and 27 years
- 360 time periods per year and 30 years

**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, 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 76  Years Tasks

<table>
<thead>
<tr>
<th>Task</th>
<th>See Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add years to the calendar.</td>
<td>“Adding Years to the Calendar” on page 338.</td>
</tr>
<tr>
<td>Add or update the description and alias for a year.</td>
<td>“Editing Year Information” on page 339.</td>
</tr>
<tr>
<td>Set the fiscal year and work with years.</td>
<td>“Setting the Fiscal Year” on page 338.</td>
</tr>
<tr>
<td>Add an All Years parent member that includes all Years members (except No Year, if that member exists).</td>
<td>“Adding or Editing Members” on page 327</td>
</tr>
</tbody>
</table>

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” on page 160. 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” on page 181.

**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 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 Smart View, 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.

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. 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” on page 340).

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 Other Options” on page 167.

To specify the currency as a reporting currency, select **Reporting Currency**. See “Working with Multiple Currencies” on page 340.

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” on page 343).

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” on page 83.

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” on page 325).
Triangulation

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.

<table>
<thead>
<tr>
<th>Task</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create a scenario.</td>
<td>See “Creating Scenarios” on page 348.</td>
</tr>
<tr>
<td>Edit a scenario.</td>
<td>See “Editing Scenarios” on page 349.</td>
</tr>
<tr>
<td>Delete a scenario.</td>
<td>See “Deleting Scenarios” on page 349.</td>
</tr>
<tr>
<td>Copy a scenario.</td>
<td>See “Copying Scenarios” on page 349.</td>
</tr>
<tr>
<td>Delete supporting detail associated with a scenario.</td>
<td>See “Deleting Supporting Detail Associated With a Scenario” on page 291.</td>
</tr>
</tbody>
</table>

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. **Optional**: For **Description**, enter a description.
6. For **Copy to Scenario**, enter a name.
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.

<table>
<thead>
<tr>
<th>Task</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create a version.</td>
<td>See “Creating Versions” on page 351.</td>
</tr>
<tr>
<td>Edit a version.</td>
<td>See “Editing a Version” on page 351.</td>
</tr>
<tr>
<td>Delete a version.</td>
<td>See “Deleting Versions” on page 352.</td>
</tr>
</tbody>
</table>

### 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. 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.
Optional: For Version, modify the name of the version.

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.

Optional: For Description, enter a description.

Optional: For Alias Table, select an alias table to associate with the version, and enter a description.

Optional: Select Enabled for Process Management to include this version in approvals.

This option is not available for target versions.

Click **Save**.

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**.
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

Subtopics

- About Aggregate Storage
- Aggregate Storage Outline Plan Type Characteristics
- Process for Adding an Aggregate Storage Database to a Planning Application
- Working with Plan Types

Planning administrators can add an aggregate storage outline using Planning Application Administration.

Note: Aggregate storage outlines are not supported using Performance Management Architect.

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 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

- Aggregate storage outlines are not supported using Performance Management Architect. This includes using Data Synchronization and any other Performance Management Architect activity.

- 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 Planning Application:

1. Create an aggregate storage plan type. Perform one task:
   - Create an aggregate storage plan type during application creation. See “Creating Applications” on page 298.
   - Add a new plan type using the plan type editor.

2. Add dimensions to the aggregate storage plan type. See “Working with Dimensions” on page 312.

   **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” on page 326.

4. Refresh the outline for the Planning application. See “Creating and Refreshing Application Databases” on page 83.

5. Create a form using the dimensions associated with the aggregate storage database. See Chapter 6, “Managing Forms.”

Working with Plan Types

You can add a plan type by selecting **Administration**, then **Manage**, and then **Plan Types**. The plan type editor enables administrators to add or delete plan types.

The number of plan types you can add will depend on which Planning modules are used.

<table>
<thead>
<tr>
<th>Application</th>
<th>Generic Block Storage Plan Types</th>
<th>Module Block Storage Plan Types</th>
<th>Aggregate Storage Plan Types</th>
<th>Total Plan Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core Planning</td>
<td>3</td>
<td>NA</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>Oracle Project Financial Planning</td>
<td>3</td>
<td>3</td>
<td>7</td>
<td>13</td>
</tr>
<tr>
<td>Capital Asset Planning</td>
<td>3</td>
<td>1</td>
<td>5</td>
<td>9</td>
</tr>
<tr>
<td>Workforce Planning</td>
<td>3</td>
<td>1</td>
<td>5</td>
<td>9</td>
</tr>
<tr>
<td>Oracle Hyperion Capital Asset Planning and Oracle Hyperion Workforce Planning</td>
<td>3</td>
<td>2</td>
<td>6</td>
<td>11</td>
</tr>
<tr>
<td>Application</td>
<td>Generic Block Storage Plan Types</td>
<td>Module Block Storage Plan Types</td>
<td>Aggregate Storage Plan Types</td>
<td>Total Plan Types</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>----------------------------------</td>
<td>--------------------------------</td>
<td>-----------------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>Oracle Hyperion Public</td>
<td>3</td>
<td>1</td>
<td>5</td>
<td>9</td>
</tr>
<tr>
<td>Sector Planning and Budgeting</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\(^1\)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 or delete a plan type using the plan type editor:
  1. From Planning, select Administration, then Manage, and then Plan Types.
  2. Perform one task:
     - To add a plan type, 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.
     - To delete a plan type, select a plan type, and then click Delete Plan Type.
  3. Click Save Plan Type.

Once the plan type is saved, you cannot modify the database type, database name, or application name. To modify a saved plan type, you must delete it and then create a new plan type with the corrected information.

### 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.

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” on page 91. 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:** Essbase considers the Period dimension in Planning as Generation 1, so take that into account when you set up Dynamic Time Series members.
Optional: Select an alias table and type an alias name. (If necessary, resize the window to view the fields.) See “Working with Alias Tables” on page 309.

Click Save.

Additional Supported Planning Application Features

Additional 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” on page 313). 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:

<table>
<thead>
<tr>
<th>Feature</th>
<th>More Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
<td>See “Setting Up Scenarios” on page 347 and “Specifying Versions” on page 350.</td>
</tr>
<tr>
<td>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).</td>
<td>See “Working with the Years Dimension” on page 338, “Editing the BegBalance Member” on page 339, and “Editing Exchange Rate Tables” on page 87.</td>
</tr>
<tr>
<td>Turn on two pass calculation at the root level, for example, for Account.</td>
<td>See “Adding or Editing Members” on page 327.</td>
</tr>
<tr>
<td><strong>Caution!</strong> 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.</td>
<td></td>
</tr>
<tr>
<td>For attributes, create hierarchies and assign aliases.</td>
<td>See “Working with Attributes” on page 332.</td>
</tr>
</tbody>
</table>

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

When using alternate hierarchies, the parent for an alternate hierarchy member can either be the root member of the Period dimension, such as Periods, or another alternate hierarchy member.
Working with Applications that Use Planning Application Administration

You can create and update applications that use Planning application administration if you are assigned the 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 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” on page 91. 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” on page 36.

➢ To refresh the application database using a utility:

1. Back up the application. See “Backing Up Applications and Application Databases” on page 91.
2. Locate the CubeRefresh utility.

   The utility is in the planning1 directory. For the full path to planning1, see “About EPM Oracle Instance” on page 53.
3. Launch CubeRefresh from the planning1 directory by entering the command and its parameters in the command line:


   • 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” on page 52.

   • 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.
- /RMI.PORT: 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, 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 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 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 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” on page 53.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[-f:passwordFile]</td>
<td>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 passwordFile. See “Suppressing Password Prompts in Planning Utilities” on page 52.</td>
</tr>
<tr>
<td>servername</td>
<td>The server name where the Planning application resides</td>
</tr>
<tr>
<td>username</td>
<td>The name of the Planning administrator</td>
</tr>
<tr>
<td>application</td>
<td>The name of the Planning application containing the dimension members to sort</td>
</tr>
<tr>
<td>member</td>
<td>The parent member whose children or descendants to sort</td>
</tr>
<tr>
<td>children</td>
<td>descendants</td>
</tr>
<tr>
<td>ascend</td>
<td>descend</td>
</tr>
</tbody>
</table>

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 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” on page 53.

Table 79  DeleteSharedDescendants Utility Parameters

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[-f:passwordFile]</td>
<td>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 passwordFile. See “Suppressing Password Prompts in Planning Utilities” on page 52.</td>
</tr>
<tr>
<td>servername</td>
<td>The server name on which the Planning application resides.</td>
</tr>
<tr>
<td>username</td>
<td>The name of the Planning administrator</td>
</tr>
<tr>
<td>application</td>
<td>The name of the Planning application containing the shared dimension members to delete.</td>
</tr>
<tr>
<td>member</td>
<td>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 member_name were found” is displayed.</td>
</tr>
</tbody>
</table>

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

Subtopics

- Working with Smart Lists
- Working with UDAs
- Working with Member Formulas

For applications that use 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 Planning application administration and a reporting application. See “Synchronizing Smart Lists in Reporting Applications” on page 364.

**Synchronizing Smart Lists in Reporting Applications**

For applications that use 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” on page 282.

1. To synchronize Smart Lists in reporting applications:
   1. Select **Administration**, then **Manage**, then **Smart Lists**.
   2. 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” on page 365.

3. 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.

**Adding or Changing Smart List Properties**

Use the Edit Smart List Properties tab to set Smart List properties.

1. 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 80  Smart List Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smart List</td>
<td>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.</td>
</tr>
<tr>
<td>Label</td>
<td>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” on page 384.</td>
</tr>
<tr>
<td>Display Order</td>
<td>How Smart Lists are sorted in the drop-down list: by ID, Name, or Label</td>
</tr>
</tbody>
</table>
| #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:  
- 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:  
- 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>
<thead>
<tr>
<th>Entry Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Unique number that sets the order for the displayed entry. Customizable only if <strong>Automatically Generate ID</strong> is not selected on the Properties tab.</td>
</tr>
<tr>
<td>Name</td>
<td>Unique alphanumeric name containing alphanumeric and underscore characters (for example: Customer_Feedback) and no special characters or spaces</td>
</tr>
<tr>
<td>Label</td>
<td>Displayed text for the Smart List entry on the drop-down list (for example: Customer Feedback).</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Option</th>
<th>Guideline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blank</td>
<td>When designing forms, select <strong>Display Missing Values as Blank</strong>. When setting Smart List properties, select <strong>Form Setting</strong>.</td>
</tr>
<tr>
<td>#MISSING</td>
<td>When designing forms, do not select <strong>Display Missing Values as Blank</strong>. When setting Smart List properties, select <strong>Form Setting</strong>.</td>
</tr>
<tr>
<td>A custom label, such as “No Change”</td>
<td>When setting Smart List properties, enter the custom label in the <strong>#MISSING Drop-Down Label</strong> field (for example, <strong>No Change</strong>). Select <strong>Drop-Down Setting</strong>.</td>
</tr>
</tbody>
</table>

**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: \(\text{UDAs: HSP\_UDF}\).

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” on page 36 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 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**

Planning member formulas support 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 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 (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

Planning formula expressions can include these predefined variables and functions.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenInputValueBlock</td>
<td>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.</td>
</tr>
<tr>
<td>CloseInputValueBlock</td>
<td>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.</td>
</tr>
<tr>
<td>NumberOfPeriodsInYear</td>
<td>Returns the number of time periods in the year</td>
</tr>
<tr>
<td>NumberOfYears</td>
<td>Returns the number of years in the application</td>
</tr>
</tbody>
</table>
## Functions in Formula Expressions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dimension(dimTag)</strong></td>
<td>Returns the name of a predefined dimension. The dimtags are:</td>
</tr>
<tr>
<td></td>
<td>• DIM_NAME_PERIOD</td>
</tr>
<tr>
<td></td>
<td>• DIM_NAME_YEAR</td>
</tr>
<tr>
<td></td>
<td>• DIM_NAME_ACCOUNT</td>
</tr>
<tr>
<td></td>
<td>• DIM_NAME_ENTITY</td>
</tr>
<tr>
<td></td>
<td>• DIM_NAME_SCENARIO</td>
</tr>
<tr>
<td></td>
<td>• DIM_NAME_VERSION</td>
</tr>
<tr>
<td></td>
<td>• DIM_NAME_CURRENCY</td>
</tr>
<tr>
<td><strong>Period(periodName)</strong></td>
<td>Returns the specified period. The periodName options are:</td>
</tr>
<tr>
<td></td>
<td>• FIRST_QTR_PERIOD</td>
</tr>
<tr>
<td></td>
<td>• SECOND_QTR_PERIOD</td>
</tr>
<tr>
<td></td>
<td>• THIRD_QTR_PERIOD</td>
</tr>
<tr>
<td></td>
<td>• FOURTH_QTR_PERIOD</td>
</tr>
<tr>
<td></td>
<td>• FIRST_PERIOD</td>
</tr>
<tr>
<td></td>
<td>• LAST_PERIOD</td>
</tr>
<tr>
<td><strong>CrossRef(accountName)</strong></td>
<td>Generates a cross reference for the account</td>
</tr>
<tr>
<td><strong>CrossRef(accountName, prefix)</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>getCalendarTPIndex()</strong></td>
<td>Generates a member formula that returns an index for the time period; the index is based on the calendar year.</td>
</tr>
<tr>
<td><strong>getFiscalTPIndex()</strong></td>
<td>Generates a member formula that returns an index for the time period; the index is based on the fiscal year.</td>
</tr>
<tr>
<td><strong>CYTD(memberName)</strong></td>
<td>Generates a calendar year-to-date formula for the member</td>
</tr>
<tr>
<td><strong>CYTD(memberName, calTPIndexName, fiscalTPIndexName)</strong></td>
<td>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.”</td>
</tr>
</tbody>
</table>

### 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 Reports

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.

<table>
<thead>
<tr>
<th>Report Type</th>
<th>Sample File Name</th>
<th>Template Name</th>
<th>Related Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form</td>
<td>PlanningFormSample.xml</td>
<td>PlanningFormTemplate.rtf</td>
<td>See &quot;Creating Simple Forms&quot; on page 159.</td>
</tr>
</tbody>
</table>
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:


   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 WEB-INF/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.

<table>
<thead>
<tr>
<th>Report Type</th>
<th>Sample File Name</th>
<th>Template Name</th>
<th>Related Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form Definition</td>
<td>PlanningFormDefSample.xml</td>
<td>PlanningFormDefTemplate.rtf</td>
<td>See “Printing Form Definitions” on page 156.</td>
</tr>
<tr>
<td>Planning Unit Annotations</td>
<td>PlanningPUAnnotationSample.xml</td>
<td>PlanningPUAnnotationTemplate.rtf</td>
<td>See “Printing Planning Unit Annotations” on page 241.</td>
</tr>
</tbody>
</table>
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 Oracle Business Intelligence 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.

<table>
<thead>
<tr>
<th>Type of Template</th>
<th>File Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>All forms</td>
<td>PlanningFormTemplate_Custom.rtf</td>
</tr>
<tr>
<td>Specific form</td>
<td>PlanningFormTemplate_Data_Form_Name</td>
</tr>
<tr>
<td></td>
<td>PlanningFormTemplate_Income.rtf</td>
</tr>
<tr>
<td>Form definition</td>
<td>PlanningFormDefTemplate_Custom.rtf</td>
</tr>
<tr>
<td>Task list status reports</td>
<td>PlanningTaskListTemplate_Custom.rtf</td>
</tr>
<tr>
<td>Planning unit annotation reports</td>
<td>PlanningPUAnnotationTemplate_Custom.rtf</td>
</tr>
</tbody>
</table>

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 HspJS.jar file at the root level (the same level as the template files).

9 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 Planning Tools page by adding links to commonly used tools for analyzing, tracking, and reporting on planning data. Planning users can then open Planning
Customizing Cascading Style Sheets

In 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” on page 379.

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>
<thead>
<tr>
<th>Customizable Sections</th>
<th>Sections Not Recommended for Customizing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-Navigation Styles</td>
<td>Tadpole Menubar</td>
</tr>
<tr>
<td></td>
<td>Tadpole MenuItem</td>
</tr>
<tr>
<td></td>
<td>Menu Styles (order is important)</td>
</tr>
<tr>
<td>Tadpole Masthead</td>
<td>Tadpole Minimized Masthead</td>
</tr>
<tr>
<td>Tadpole Content Area</td>
<td>Tadpole Toolbar</td>
</tr>
<tr>
<td>Tadpole Logon Styles</td>
<td>Tadpole Viewpane</td>
</tr>
<tr>
<td>Tabs</td>
<td>Tadpole Tree</td>
</tr>
<tr>
<td>Tab Anchor Tags</td>
<td></td>
</tr>
<tr>
<td>Tab Body</td>
<td></td>
</tr>
<tr>
<td>TaskTabs</td>
<td></td>
</tr>
<tr>
<td>Task Tab Body</td>
<td></td>
</tr>
</tbody>
</table>
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” on page 379.

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” on page 381.

### Location of Cascading Style Sheet Files

The cascading style sheets for 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

Subtopics

- Changing the Color of the Content Area Background
- Changing the Color of Hyperlinks and Visited Hyperlinks

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 85  Changing the Color of the Content Area Background

<table>
<thead>
<tr>
<th>Default Content Area Background</th>
<th>White Content Area Background</th>
</tr>
</thead>
<tbody>
<tr>
<td>.content table.content {</td>
<td>.content table.content {</td>
</tr>
<tr>
<td>background: #e5eaef;</td>
<td>background: #ffffff;</td>
</tr>
<tr>
<td>}</td>
<td>}</td>
</tr>
</tbody>
</table>

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 86  Changing the Color of Hyperlinks and Visited Hyperlinks

<table>
<thead>
<tr>
<th>Black Hyperlinks and Visited Hyperlinks</th>
<th>Red Hyperlinks and Visited Hyperlinks</th>
</tr>
</thead>
<tbody>
<tr>
<td>a:link, a:visited {</td>
<td>.content table.content {</td>
</tr>
<tr>
<td>color: #000000;</td>
<td>background: #ff0000;</td>
</tr>
<tr>
<td>}</td>
<td>}</td>
</tr>
</tbody>
</table>
Customizing the Style of Row and Column Members in Forms

You can customize 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 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” on page 378.**

2. **In the planning.css file, customize these header tags:**

   Table 87   Header Tags in Planning.css File
   
<table>
<thead>
<tr>
<th>Rows</th>
<th>Columns</th>
</tr>
</thead>
<tbody>
<tr>
<td>rowHeader_0</td>
<td>columnHeader_0</td>
</tr>
<tr>
<td>rowHeader_1</td>
<td>columnHeader_1</td>
</tr>
<tr>
<td>rowHeader_2</td>
<td>columnHeader_2</td>
</tr>
<tr>
<td>rowHeader_3</td>
<td>columnHeader_3</td>
</tr>
<tr>
<td>rowHeader_4</td>
<td>columnHeader_4</td>
</tr>
</tbody>
</table>

   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 the Style of Header Cells in Forms

Applying Bolding to All Forms

To apply a bold style to all forms:

1. From HyperionPlanning.war\custom, open HspCustom.css.
2. Insert this class code:
   
   ```
   customheaderStyleFont-family:<font>;Font-size:<size>;Font-weight:bold;
   ```
3. You can also change the font and the font size in the same portion of code.
   
   For example, to use bolded garamond 14 as the style for header cells you would specify:
   
   ```
   customheaderStyleFont-family:garamond;Font-size:14;Font-weight:bold;
   ```
4. Perform the tasks in “Registering Customized Skins” on page 383.

Applying Bolding to One Form

To apply bolding to a single form:

1. Determine the ID of the form to customize by running this query in which <FORM_NAME> is the name of the form:
   
   ```
   Select object_id AS FORM_ID,object_name AS FORM_NAME from hsp_object where object_name like '<FORM_NAME>'
   ```
   
   For example, to modify a form called "1.30 Per Payor – Metrics", you would run this query:
   
   ```
   Select object_id AS FORM_ID,object_name AS FORM_NAME from hsp_object where object_name like '1.30 Per Payor-Metrics'
   ```
2. When the ID is returned, create a css class in HspCustom.css using the following format.
   
   ```
   "customheaderStyle_<FORM_ID>"
   ```
3. Set font-weight in customheaderStyle as bold as shown below. You can also change the font and the font size in the same portion of code.
   
   ```
   customheaderStyleFont-family:customheaderStyleFont-family:<font>;Font-size:<size>;Font-weight:bold;
   ```
4. To use different styles on individual forms, create a css class for each form.
5. Perform the tasks in “Registering Customized Skins” on page 383.

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:
   
   ```
   ```
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” on page 383.

## 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/e15862/toc.html](http://docs.oracle.com/cd/E23943_01/apirefs.1111/e15862/toc.html)
About Customizing Text, Color, and Images

You can customize text, colors, and images in the Planning Web interface, and you can localize text.

- “Customizing Text, Color, and Images” on page 384
- “Customizing Text in Non-Latin Languages” on page 385
- “Restricted Characters in Customized Messages” on page 387

Customizing Text, Color, and Images

You can customize text, color, and images for these items in the 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” on page 387.
- 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” on page 378.
- 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\n   ```

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” on page 191.

### Customizing Text in Non-Latin Languages

You can customize text in the 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.

1 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 is as follows:

   native2ascii [-encoding language_encoding] [inputfile_name[outputfile_name]]

<table>
<thead>
<tr>
<th>Language</th>
<th>Encoding Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Russian</td>
<td>Cp1251</td>
</tr>
<tr>
<td>Turkish</td>
<td>Cp1254</td>
</tr>
<tr>
<td>Japanese</td>
<td>SJIS</td>
</tr>
<tr>
<td>Simplified Chinese</td>
<td>EUC_CN</td>
</tr>
<tr>
<td>Traditional Chinese</td>
<td>Big5</td>
</tr>
<tr>
<td>Korean</td>
<td>EUC_KR</td>
</tr>
</tbody>
</table>

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.
 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:

```javascript
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” on page 387
- “Customizing States” on page 388
- “Customizing Actions” on page 388

 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:

```sql
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:

```sql
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:

```sql
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:

```sql
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 server 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 89  HSP_SPREAD_PATTERN Table

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME — VARCHAR(80)</td>
<td>Internal name (not displayed)</td>
</tr>
<tr>
<td>UNIQUE NOT NULL</td>
<td></td>
</tr>
<tr>
<td>LABEL — VARCHAR(80)</td>
<td>Reference to a string ID in the resource file, which displays in the user</td>
</tr>
<tr>
<td>NOT NULL</td>
<td>interface. If a string resource is not created, the LABEL string identifier</td>
</tr>
<tr>
<td>PATTERN VARCHAR(255)</td>
<td>Space delimited; one or more elements, up to 255 characters:</td>
</tr>
<tr>
<td>NOT NULL</td>
<td>• A number: For example, to specify that the parent value splits evenly</td>
</tr>
<tr>
<td></td>
<td>across its children, specify 1.</td>
</tr>
<tr>
<td></td>
<td>• !: Lock; do not change the value of a cell that has “!” as its pattern</td>
</tr>
<tr>
<td></td>
<td>element.</td>
</tr>
<tr>
<td></td>
<td>• P: Parent value being spread. A child receives the exact value the parent</td>
</tr>
<tr>
<td></td>
<td>receives.</td>
</tr>
<tr>
<td></td>
<td>• <em>: Repeat the element that follows. Use 1</em> to repeat 1 for every cell, to</td>
</tr>
<tr>
<td></td>
<td>take up the rest of the space in the applied pattern. If no * exists as</td>
</tr>
<tr>
<td></td>
<td>part of an element within a pattern, the entire pattern repeats itself</td>
</tr>
<tr>
<td></td>
<td>instead of a specific element. A pattern cannot contain only the *</td>
</tr>
<tr>
<td></td>
<td>character, and can have only one element with the * character.</td>
</tr>
<tr>
<td></td>
<td>For example, 1 2* 3 is allowed, but 1 2* 3* is not.</td>
</tr>
</tbody>
</table>

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” on page 390.

### 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 90  Examples of Applied Spreading Patterns

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Cell 1</th>
<th>Cell 2</th>
<th>Cell 3</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 4 5</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>The new value is spread to level zero members using the 4 4 5 pattern. For</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>example, if you set Q1 to 13, its value is spread as:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Jan = 4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Feb = 4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mar = 5</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1 is the same as 1*. This FLOW spread type shows how FLOW works if no</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>values exist to make the spread proportional. The parent’s value is</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>divided by 3, the number of children cells, and spread equally to each of</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>its three children.</td>
</tr>
<tr>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>Each child cell receives the parent’s value. This is more a copy pattern</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>than a spread.</td>
</tr>
</tbody>
</table>

390  Customizing Planning Web Client
Creating Instructions for Planners

Using 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” on page 171.

- 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” on page 160), and then link cells to documents in EPM 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” on page 182.

- 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” on page 82.
In This Chapter

Calculation Script is Too Long or Complex ............................................................ 394
Cannot Find Member .......................................................................................... 394
Cannot Process Request ................................................................................... 395
Conflicting Change by Users ............................................................................. 395
Copy Version Error .......................................................................................... 395
Currency Conversion Calc Script Failed ............................................................ 396
Form Error Message ......................................................................................... 396
Database Locked by Another Administrator ....................................................... 397
FDM Drill Through Issues ................................................................................. 397
500 Error Message .......................................................................................... 398
Implied Sharing Issues .................................................................................... 398
Invalid Value When Entering Date Value ......................................................... 399
Manage Database Errors .................................................................................. 400
Maximum Number of Applications Already Connected or Database is Locked 401
Optimizing WebLogic Server Parameters ....................................................... 401
Optimizing Windows Network Parameters ...................................................... 402
Registering a Restored Application with Planning ............................................ 403
Session Timeout .................................................................................................. 403
Slow Performance ............................................................................................ 404
Slow Performance When Opening Forms Using a Dial-Up Connection .............. 404
Slow Performance When Opening Large Forms ................................................ 405
Unable to Create Application Error ................................................................. 406
Unable to Refresh Application Error ................................................................. 406
Unsuccessful Log On ........................................................................................ 407

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 calc scripts that you can launch from the Web to convert currencies. When generating calc scripts, if the following error displays, the calc 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 calc script.

Solution:

➢ To resolve this issue:

1. Limit the calc script to one scenario.
   - If the calc script generates successfully, skip to the last step.

2. If the error still displays, limit the calc script to one reporting currency.
   - If the calc 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 calc 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 calc 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 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 calc script command and a comma for each member name. Assuming a calc script command length of 500 characters, you can use this formula:

\[(\text{Number of members} \times \text{Average length of member names}) \text{ + Number of members} + 500 \leq 64,000\]

**Currency Conversion Calc 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 calc 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 Planning after an 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, Planning applications may become locked. This can be caused by events such as abnormally exiting the application.

Solution:
See “Unlocking Applications” on page 81.

FDM Drill Through Issues

Scenario:
If you are using FDM for drill through, and you deploy FDM using a non-standard context path, users cannot drill through to FDM in forms.

Solution:
Specify the new FDM path in a Planning application property. The context path is the part of the URL that accesses the deployed Web application. For example, in this URL, HyperionFDM is the context path: http://host:port/HyperionFDM/AuthorizedPages/. If you are using the standard context path of HyperionFDM, you do not need to add or change this property.

To resolve FDM drill-through issues if FDM is deployed to a non-standard context path:

1. In the Planning application, select Administration, then Manage Properties.
2. Select Application Properties.
3. Click Add, and enter this property name (avoid using spaces):
   ADD_HSP_RATES_DTURL_PATTERN

   The value for this property is a regular expression enclosed by slashes and asterisks: /* */. For example, the default value for FDM is /*HyperionFDM*/. To specify a different context path, substitute the new information for HyperionFDM.

4. In the blank row, enter the new context path for Oracle Hyperion Financial Data Quality Management, enclosed by a forward slash and asterisk /* */, as in this hypothetical example:
   /*NewPath*/

5. Click Save and confirm your changes.
6. Stop and then restart the application server.
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 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 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.

Invalid Value When Entering Date Value

Scenario:

When trying to enter a value into a form cell, you get the error, “You have entered an invalid value. Please try again.” For example, you try to enter a formatted date value (for example, 12/8/2010) into a cell whose Data Type is Date and get this error.

Solution:

Depending on the type of data users want to enter in a cell, in Oracle Hyperion EPM Architect, set the dimension of the member with this data type as the first in the Data Type Evaluation Order.
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. 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 Planning server before making changes.

➢ To check the environment variables for Windows 2000:

2. Close Essbase.
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.
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:\temp.

   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” on page 89.
- Your application uses WebLogic Server.
Solution:

To tune WebLogic Server performance settings:

1. Open the WebLogic Server Administration Console.
2. Select the deployed 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 Planning application, but it is not available for selecting in EPM Workspace.

**Solution:**

Register the application with 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 Oracle Hyperion Enterprise Performance Management 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**

**Scenario:**
Performance is generally slow.

**Solution:**
Ensure that you are using the recommended configuration for optimizing performance described in the Planning Release 11.1.2.2 *Oracle Hyperion Planning New Features* and Readme.

**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 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-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.

**Slow Performance When Opening Large Forms**

**Scenario:**

Unusually large forms with many cells may require significant time to open.
Solution:

You can display a warning when users try to open forms that are larger than a specified number of cells. The default value for the number of cells that trigger the warning is 5,000. Administrators can change the default value.

To set the number of form cells that trigger a warning message upon opening:

1. In Planning, select File, then Preferences.
2. Select Display Options.
3. In Warn If Form Larger Than Cells Specified, enter a value.

Unable to Create Application Error

Scenario:
When you try to create an application in 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:

- 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, an 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 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 Planning Web client, this error displays: “Unsuccessful login. Please try again.”

Solutions:

Ensure that you are using a valid username and password.

Ensure that 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.
About Customizing Forms

Note: All of the customization information in this section is applicable only if the application is using the Planning Release 11.1.2.1 user interface and features, as described in “Using the Planning Release 11.1.2.1 User Interface and Features” on page 48.

This section describes using JavaScript to customize forms. If you are a programmer familiar with JavaScript, you can customize the look and behavior of Planning forms by writing custom JavaScript and including it in applications.

You can customize forms in these ways:

- Add buttons to forms that link to Web pages or run custom JavaScript code.
- Modify application behavior when users save forms.
- Modify application behavior when the focus enters or leaves a cell.

There are no additional requirements for users when including custom JavaScript in your Planning application. Administrators need not take special steps to enable custom JavaScript for applications.

Modifying JavaScript

To add custom JavaScript to a Planning application, modify the code in ValidateData.js, which is located in the \custom subdirectory wherever the .war file is extracted. For examples of customizing forms, see the sample code and comments in SampleValidateData.js. (All of the customization information in this section is applicable only if the application is using the Planning Release 11.1.2.1 user interface and features, as described in “Using the Planning Release 11.1.2.1 User Interface and Features” on page 48.)
**customCellEnterPre**

**Description**

Use `customCellEnterPre` to change the behavior when users click in cells, before default Planning logic runs. For example, when users click in cells, a message can indicate the acceptable range of values to enter. When users click in certain types of cells, such as accounts, a message can prompt users to enter supporting detail.

After `customCellEnterPre` runs, if it returns True, the default Planning logic runs. If it returns False, no additional logic follows the custom code. `SampleValidateData.js` includes an example that displays a message prompting users to enter supporting detail when clicking in certain cells.

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Row</td>
<td>An integer that defines the row for the cell.</td>
</tr>
<tr>
<td>Col</td>
<td>An integer that defines the column for the cell.</td>
</tr>
<tr>
<td>Cell</td>
<td>A handle for the HTML input artifact.</td>
</tr>
</tbody>
</table>

**Returns**

<table>
<thead>
<tr>
<th>Return Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>The Planning default logic runs after this code completes.</td>
</tr>
<tr>
<td>False</td>
<td>No further logic (Planning default logic or <code>customCellEnterPost</code>) runs after this completes.</td>
</tr>
</tbody>
</table>

**See Also**

`customCellEnterPost`

**customCellEnterPost**

**Description**

Use `customCellEnterPost` for custom behavior when focus comes into a cell, after the default Planning logic runs. This code runs when users click in a cell, after the Planning default logic if `customCellEnterPre` and the default logic return True. If they return False, this function is disabled.

`SampleValidateData.js` includes an example for `customCellEnterPre` that you can use for `customCellEnterPost`. The example displays a message prompting the user to enter supporting detail clicking in specific cells.
Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Row</td>
<td>An integer that defines the row for the cell.</td>
</tr>
<tr>
<td>Col</td>
<td>An integer that defines the column for the cell.</td>
</tr>
<tr>
<td>Cell</td>
<td>A handle for the HTML input artifact.</td>
</tr>
</tbody>
</table>

Returns

<table>
<thead>
<tr>
<th>Return Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>The Planning default logic runs after this code completes.</td>
</tr>
<tr>
<td>False</td>
<td>No further logic runs after this completes.</td>
</tr>
</tbody>
</table>

See Also

customCellEnterPre

customCellValidatePre

Description

Use `customCellValidatePre` for custom behavior when focus leaves a cell, before default Planning logic runs. For example, you can compare the value a user entered for the cell to another value and display an alert if the variance is too great or out of range. You can define a spreading algorithm that occurs when the focus leaves a cell, replacing the default spreading algorithm. For example, Planning usually spreads by time period, but you can write custom JavaScript that pre-aggregates the data on the client side before writing values back to Essbase.

After this function runs, if `customCellValidatePre` returns True, Planning default logic runs. If it returns False, no additional logic follows the custom code. Forms use your custom code and skip the default Planning logic, which could cause users to submit invalid data.

`SampleValidateData.js` includes an example that executes additional validation when the focus leaves a cell. When the focus leaves a cell, the code compares the value the user entered for the current year to last year’s value. If the current year value is over 10% larger than the previous year, a message is displayed.

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Row</td>
<td>An integer that defines the row for the cell.</td>
</tr>
<tr>
<td>Col</td>
<td>An integer that defines the column for the cell.</td>
</tr>
</tbody>
</table>
**Argument** Description

*Cell* A handle for the HTML input artifact.

**Returns**

**Argument** Description

*True* The Planning default logic runs after this code completes.

*False* No further logic runs after this code completes.

**See Also**
customCellValidatePost

customCellValidatePost

**Description**

Use customCellValidatePost for custom behavior when focus leaves a cell, after the default Planning logic runs. This is similar to customCellValidatePre, which runs when the focus leaves a cell, after the Planning default logic if this function and the Planning default logic return True. If they return False, this function is disabled. Return values have no effect.

SampleValidateData.js includes an example for customCellValidatePre that you can use for this code. The example executes additional validation when the focus leaves a cell. The JavaScript code compares the value entered for the current year to the value for last year. If the current year value is over 10% larger than the previous year value, a message is displayed.

**Arguments**

**Argument** Description

*Row* An integer that defines the row for the cell.

*Col* An integer that defines the column for the cell.

*Cell* A handle for the HTML input artifact.

**Returns**

**Return Value** Description

*True* The Planning default logic runs after this code completes.

*False* No further logic runs after this completes.

**See Also**
customCellValidatePre
**customOnLoad**

**Description**

Use `customOnLoad` for custom behavior when forms are loaded or reloaded. A JavaScript variable called `savedData` indicates whether Save has been completed. For example, when users open forms, you could display instructions or determine if they match tolerances and inform users of any corrective actions.

**Arguments**

No arguments.

**Returns**

No return values.

**drawCustomButtons**

**Description**

Use `drawCustomButton` to add custom buttons to forms. Buttons can link to any data entry page or launch custom JavaScript code included in `ValidateData.js`. For example, you can add buttons to validate forms or run reports.

`SampleValidateData.js` includes an example that adds a button labeled `Validate` to a form. When a user clicks the button, the JavaScript code compares the values entered for the current year to last year's values. If the current year's values are more than 10% larger than the previous year's values, a message is displayed.

**Arguments**

No arguments.

**Returns**

No return values.

**validateForm**

**Description**

Use `validateForm` to provide form-level behavior that is launched when users click Save. With default Planning behavior, when users click Save on forms, the `validateForm` function submits the grid. For example, you can calculate a variance between budget and actuals by comparing values in two columns, displaying an alert when users click Save if the variance is too high.
SampleValidateData.js includes an example that executes additional validation when users click Save. The values entered for the current year are compared to last year’s values. If the current year values are more than 10% larger than previous year values, a message displays.

**Arguments**

No arguments.

**Returns**

<table>
<thead>
<tr>
<th>Return Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>Saves the grid.</td>
</tr>
<tr>
<td>False</td>
<td>Cancels the save.</td>
</tr>
</tbody>
</table>

**Deploying Custom JavaScript**

When using custom JavaScript in Planning applications:

- Specify which forms the code applies to, as shown in SampleValidateData.js.
- Maintain custom JavaScripts when upgrading Planning. When upgrading or modifying standard Planning JavaScript files, you must merge your code with the updated version of ValidateData.js. (If there are no changes to ValidateData.js, you can back up the file before upgrading Planning and copy your version of the file.)
- The default Planning calc scripts can run after forms are saved, so default calc scripts could overwrite the actions of custom JavaScript. If custom calc scripts run automatically after forms are saved, calc scripts could overwrite actions of custom JavaScripts. If custom JavaScript determines how values are propagated throughout the hierarchy, be aware of possible conflicts with calc scripts.
- Custom JavaScript can modify cells that are not visible on forms. For example, if a quarter is collapsed, custom JavaScript can still affect values for individual months.
- Custom JavaScript cannot affect suppressed rows or columns.
- Depending on the complexity of code, cell-level JavaScript functions can have an adverse effect on application performance. For complex JavaScript, consider using the form-level function (validateForm) or custom buttons to launch JavaScript code.
- Utility functions are included in validateDataHelper.js.
- If custom JavaScript overrides default Planning behavior, you are responsible for ensuring that data is valid. You can verify data after running custom JavaScript using enterData.js. For examples, see LeaveCell.

All of the customization information in this section is applicable only if the application is using the Planning Release 11.1.2.1 user interface and features, as described in “Using the Planning Release 11.1.2.1 User Interface and Features” on page 48.
In This Appendix

Restrictions for Applications and Databases .......................................................... 415
Restrictions for Dimensions, Members, Aliases, and Forms ......................................... 416
Dimension and Member Names in Calc Scripts, Report Scripts, Formulas, Filters, and Substitution Variables ................................................................. 417
Restrictions for User Names ............................................................................ 418

Restrictions for Applications and Databases

When naming an application, follow these rules:

- For non-Unicode mode applications and databases, use no more than 8 bytes; for Unicode-mode applications and databases, use no more than 30 characters.
- 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
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 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 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.
- These words:

  ALL AND ASSIGN AVERAGE CALC
  CALCMBR COPYFORWARD CROSSDIM
  CURMBRNAME DIM DIMNAME DIV
  DYNAMIC EMPTYPARM EQ EQOP EXCEPT
  EXP EXPERROR FLOAT FUNCTION GE
  GENRANGE GROUP GT ID IDERROR INTEGER
  LOOPOPARGS LT MBR MBRNAME MBRONLY
  MINUS MISSING MUL MULOP NE
  NONINPUT NOT OR PAREN PARENPARMS
  PERCENT PLUS RELOP SET SKIPBOTH SKIPMISSING
  SKIPNONE SKIPZERO TO
  TOLOCALRATE TRAILMISSING
  TRAILSUM UMINUS UPPER
  VARORXMBR XMBRONLY $
  \$UNIVERSE\$
  #MISSING
  #MI

- If Dynamic Time Series is enabled, do not use History, Year, Season, Period, Quarter, Month, Week, or Day.

**Dimension and Member Names in Calc Scripts, Report Scripts, Formulas, Filters, and Substitution Variables**

In substitution variable values, calc 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:

  &  ampersand
  *  asterisk
  @  at sign
  \  backslash
  { }  braces
  :  colon
  ,  comma
  -  dash, hyphen, or minus
  !  exclamation point
  =  equal sign
  >  greater than sign
  <  less than sign
  ()  parentheses
  %  percent sign
  .  period
  +  plus sign
  ;  semicolon
  /  slash
  ~  tilde
In calculation scripts and formulas, enclose member names that are also 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 quotes names that contain these characters in calc scripts, report scripts or formulas, and names that start with these characters: Spaces + - * / ( ) : , @ ; ) { } [ ] <

**Restrictions for User Names**

User names can have up to 80 characters.
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” on page 420.

Note: 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 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” on page 420.
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.
On Layout, select a numbered formula row or column.

Use Segment Properties to modify the formula properties.

Click to change the operation or function performed.

Save the form.

Deleting Formulas

To delete a formula:

1 Select the appropriate 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” on page 169.

The syntax for mathematical functions is:

FunctionName(arguments)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FunctionName</td>
<td>The name of a mathematical function.</td>
</tr>
<tr>
<td>arguments</td>
<td>A numeric value, a row, column, or cell reference, or an embedded function.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abs</td>
<td>Returns the absolute value of numeric values or references.</td>
</tr>
<tr>
<td>Average</td>
<td>Returns the average of a group of numeric values or references.</td>
</tr>
<tr>
<td>AverageA</td>
<td>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.</td>
</tr>
</tbody>
</table>
### Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Returns the number of values in a group of numeric values or references.</td>
</tr>
<tr>
<td>CountA</td>
<td>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.</td>
</tr>
<tr>
<td>Difference</td>
<td>Returns the absolute value of a numeric value or reference subtracted from another numeric value or reference.</td>
</tr>
<tr>
<td>Eval</td>
<td>Evaluates an expression. Eval is useful for embedding expressions as function arguments.</td>
</tr>
<tr>
<td>IfThen, If</td>
<td>Returns one value if a condition equals true, and another value if a specified condition equals false.</td>
</tr>
<tr>
<td>Max</td>
<td>Returns the maximum value from a group of numeric values or references.</td>
</tr>
<tr>
<td>Min</td>
<td>Returns the minimum value from a group of numeric values or references.</td>
</tr>
<tr>
<td>Mod</td>
<td>Returns the remainder, modulus, from a division formula.</td>
</tr>
<tr>
<td>PercentOfTotal</td>
<td>Returns the result of a numeric value or reference divided by another numeric value or reference, multiplied by 100.</td>
</tr>
<tr>
<td>Pi</td>
<td>Returns the number 3.14159265358979, to 15 digits.</td>
</tr>
<tr>
<td>Product</td>
<td>Multiplies all numbers or references and returns the product.</td>
</tr>
<tr>
<td>Random</td>
<td>Returns a random number between 0.0 and 1.0.</td>
</tr>
<tr>
<td>Rank</td>
<td>Returns the highest or lowest value of a specified column or row.</td>
</tr>
<tr>
<td>Round</td>
<td>Rounds a number up or down by specified digits.</td>
</tr>
<tr>
<td>Sqrt</td>
<td>Returns the square root of a numeric value, row, column, or cell.</td>
</tr>
<tr>
<td>Sum</td>
<td>Returns the sum of a group of numeric values or references.</td>
</tr>
<tr>
<td>Truncate / Trunc</td>
<td>Removes the specified number of digits from numeric values.</td>
</tr>
<tr>
<td>Variance / Var</td>
<td>Evaluates the difference between the specified values based on the account type for the current account.</td>
</tr>
<tr>
<td>VariancePercent / VarPer</td>
<td>Evaluates the percentage difference between the specified values based on account type for the current account.</td>
</tr>
</tbody>
</table>

### 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

\[(numeral_1, numeral_2, \ldots, numeral_n)\]

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 is:

\[\text{FunctionName}(\text{GridName}.\text{GridElement}[\text{segment(range)}].\text{Property})\]

**Table 93: Argument Components**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GridName</strong></td>
<td>The form name. For example: Difference (grid1.row[5], grid1.row[6]) returns the difference of two rows on form grid1. Optional. If GridName is not specified, the default is the name of the current form.</td>
</tr>
<tr>
<td><strong>GridElement</strong></td>
<td>One of the following keywords: row, col, column, or cell. For example, Max(row[1], row[2], row[3]) returns the maximum value of three rows. GridElement is optional. However, a cell reference requires row and column segment identifiers. For example, cell[2, A] and [2, A] both refer to the cell that is the intersection between row 2 and column A. The keyword cell is optional. Cell references can use the [row, col] syntax or [col, row] syntax. Optional. If GridElement is not specified, letters represent columns and numbers represent rows; for example: Max ([1, A], [2, A], [3, A]) refers to rows 1, 2 and 3 of column A.</td>
</tr>
<tr>
<td><strong>segment</strong></td>
<td>A row, column, or cell reference number. For an expanded row or column, you must specify the segment. For example, row[2] addresses row segment 2. Segments are enclosed in square brackets [ ]. Required.</td>
</tr>
<tr>
<td><strong>range</strong></td>
<td>The rows, columns, or cell that are expanded from the specified segment. If range is specified, the system calculates the formula using only the specified range. For example, row<a href="">2</a> uses only the 3rd through 5th rows of expanded segment 2. Optional. When range is not provided, all expanded cells are used. <strong>Note:</strong> If a segment expands to only one row or column, do not use the range argument.</td>
</tr>
<tr>
<td><strong>property</strong></td>
<td>One of these keywords: average, averageA, count, countA, max, min, product, or sum. The property specifies how to aggregate the specified expanded rows, columns, or cells. Oracle recommends that property is not 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: Average(row[1], row[2]) In contrast, the following example first calculates the average of row[1], then the average of row[2], adds these two results, then divides by 2: Average(row[1].average, row[2].average) The default property for a row, column, or cell reference is sum. For example, row[2] is equivalent to Sum(row[2]).</td>
</tr>
</tbody>
</table>
Because segment is the only required part of a reference, the following references are the same:

\[
\text{Grid1.row[1].sum}
\]

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):

\[
\text{row[1].CountA}
\]

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:

\[
\text{row[1].Count}
\]

**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” on page 423)
- Reference Property Argument (see “Reference Property Argument” on page 424)

**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:

\[
\text{FunctionName(GridName.Gridelement[segment(range)].property)}
\]

You apply the following aggregate properties to a row, column, or cell reference.

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>Returns the average of a row, column, or cell. The calculation excludes #MISSING and #ERROR values.</td>
</tr>
<tr>
<td>AverageA</td>
<td>Returns the average of a row, column, or cell. The calculation includes #MISSING and #ERROR values.</td>
</tr>
<tr>
<td>Count</td>
<td>Returns the number of values in a row, column, or cell. The calculation excludes #MISSING and #ERROR values.</td>
</tr>
<tr>
<td>CountA</td>
<td>Returns the number of values in a row, column, or cell. The calculation treats #MISSING and #ERROR values as zero (0).</td>
</tr>
<tr>
<td>Max</td>
<td>Returns the maximum value of a row, column, or cell.</td>
</tr>
<tr>
<td>Min</td>
<td>Returns the minimum value of a row, column, or cell.</td>
</tr>
<tr>
<td>Product</td>
<td>Returns the product of rows or columns.</td>
</tr>
<tr>
<td>Properties</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Sum</td>
<td>Returns the sum of a row, column, or cell.</td>
</tr>
</tbody>
</table>

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:

\[
\text{Average}(\text{row}[2])
\]

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:

\[
\text{row}[2]
\]

**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 is:

\[
\text{AXIS[segment(range)].IfNonNumber(arg).AggregateProperty}
\]

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AXIS</td>
<td>One of these keywords: row, column, or cell. Optional.</td>
</tr>
<tr>
<td>Segment(range)</td>
<td>Indicates any valid axis reference, such as a row number, column letter.</td>
</tr>
<tr>
<td>IfNonNumber</td>
<td>Indicates how to treat missing or error data within the Axis Ref.</td>
</tr>
<tr>
<td>(arg)</td>
<td>Indicates what number to use if missing or error data is encountered within the AxisRef.</td>
</tr>
<tr>
<td>AggregateProperty</td>
<td>The aggregate function is used for aggregate segments. Optional.</td>
</tr>
</tbody>
</table>

For example:

If cell[1,A] = 3 and cell[1,B] = #MISSING,

The expression:

\[
\text{cell}[1,A] / \text{cell}[1,B]
\]

returns #ERROR.

The expression:

\[
\text{cell}[1,A] / \text{cell}[1,B].\text{IfNonnumber}(1)
\]

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:

```plaintext
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 is:

```
Abs (argument)
```

where *argument* is one of the following:

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>numeric</td>
<td>A numeric value. For example, Abs (-20) returns the value 20. Numeric values can include decimals and negative values.</td>
</tr>
<tr>
<td>row, column, or cell reference</td>
<td>A pointer to a row, column, or cell within a grid. References can be specified in several ways. The reference syntax is: GridName.GridElement[segment(range)].Property.</td>
</tr>
<tr>
<td>function</td>
<td>An embedded function.</td>
</tr>
</tbody>
</table>

**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 is:

\[ \text{Average}(\text{arguments}) \text{ or } \text{Avg}(\text{arguments}) \]

where arguments is one or more of the following:

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>numeric</td>
<td>A numeric value. For example, Average (10, 20, 30) returns the value 20. Numeric values can include decimals and negative values.</td>
</tr>
<tr>
<td>row, column, or cell reference</td>
<td>A pointer to a row, column, or cell within a grid. References can be specified in several ways. The reference syntax is: 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.</td>
</tr>
<tr>
<td>function</td>
<td>An embedded function.</td>
</tr>
</tbody>
</table>

**Examples:**

The following expression returns the value 20:

\[ \text{Avg}(10, 30, 20) \]

The following example returns the average of all numbers that are part of three aggregate rows:

\[ \text{Average}(\text{row}[1], \text{row}[6], \text{row}[8]) \]

The following example calculates the average of three aggregate columns; E, G, and I. The calculation produces three numbers, then calculates the average of the three numbers:

\[ \text{Avg}(\text{column}[E].\text{avg}, \text{column}[G].\text{avg}, \text{column}[I].\text{avg}) \]

The following example calculates the average of aggregate row 3 and divides the average by 100:

\[ \text{Avg}(\text{row}[3])/100 \]

**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 is:

\[
\text{AverageA}(\text{arguments}) \text{ or } \text{AvgA}(\text{arguments})
\]

where `arguments` is one or more of these arguments:

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>numeric</td>
<td>A numeric value. For example, AverageA (10, 20, 30) returns the value 20. Numeric values can include decimals and negative values.</td>
</tr>
<tr>
<td>row, column, or cell reference</td>
<td>A pointer to a row, column, or cell. References can be specified in several ways. The reference syntax is: 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.</td>
</tr>
<tr>
<td>function</td>
<td>An embedded function.</td>
</tr>
</tbody>
</table>

**Example**

In the following example, if a grid has 4 rows with the values 10, 20, 30, and, #ERROR. The following formula in the 5th row returns the value 15:

\[
\text{AverageA([1:4])}
\]

**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 is:

\[
\text{Count}(\text{arguments})
\]

where `arguments` is one or more of these arguments:

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>numeric</td>
<td>A numeric value. For example, Count (10, 20, 30) returns the value 3. Numeric values can include decimals and negative values.</td>
</tr>
<tr>
<td>row, column, or cell reference</td>
<td>A pointer to a row, column, or cell within a grid. References can be specified in several ways. The reference syntax is: GridName.GridElement[segment(range)].Property.</td>
</tr>
<tr>
<td>function</td>
<td>An embedded function.</td>
</tr>
</tbody>
</table>

**Examples:**

The following example returns the count of three rows, 1, 6, and 8:

\[
\text{Count(row[1], row[6], row[8])}
\]
The following example returns the count of 3 columns:

\[
\text{Count}(\text{column}[E], \text{column}[G], \text{column}[I])
\]

The following example calculates the count of the cell located at row 4, column D:

\[
\text{Count}(\text{cell}[D,4])
\]

The following example calculates the count of aggregate row 3 in grid 5:

\[
\text{Count}(\text{grid1}.\text{row}[3])
\]

**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 is:

\[
\text{CountA}(\text{arguments})
\]

where *arguments* is one or more of these arguments:

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>numeric</td>
<td>A numeric value. For example, \text{CountA}(10,20,30,50) returns the value 4. Numeric values can include decimals and negative values.</td>
</tr>
<tr>
<td>row, column, or cell</td>
<td>A pointer to a row, column, or cell within a grid. References can be specified in several ways. The reference syntax is: \text{CountA(GridName.GridElement[segment(range)]).property}</td>
</tr>
<tr>
<td>reference</td>
<td></td>
</tr>
<tr>
<td>function</td>
<td>An embedded function.</td>
</tr>
</tbody>
</table>

**Example**

In the following example, if a grid has 4 rows with the values 10, 20, 30, and, #ERROR. The following formula in the 5th row returns the count of four rows:

\[
\text{CountA([1:4])}
\]

The following example returns the count of four rows:

\[
\text{CountA(row}[1], \text{row}[6], \text{row}[8] \text{ row[where data yields #ERROR])}
\]

**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 is:

\[
\text{Difference}(\text{arg1}, \text{arg2})
\]

where \text{arg2} is subtracted from \text{arg1} and are one or more of the following arguments:
<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>numeric</td>
<td>A numeric value. For example, Difference (3, 5) returns the absolute value 2. Numeric values can include decimals and negative values.</td>
</tr>
<tr>
<td>row, column, or</td>
<td>A pointer to a row, column, or cell within a grid. References can be specified in several ways. The reference syntax is: GridName.GridElement[segment(range)].Property.</td>
</tr>
<tr>
<td>reference</td>
<td>The following example returns the difference of two rows in form grid1: Difference( grid1.row[1], grid1.row[6] )</td>
</tr>
<tr>
<td>function</td>
<td>An embedded function.</td>
</tr>
</tbody>
</table>

**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:

\[
\text{Difference}(3, -5)
\]

The following example calculates the difference of two aggregate columns:

\[
\text{Difference}( \text{column}[E], \text{column}[G] )
\]

**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 is:

\[
\text{Eval}(\text{expression})
\]

where expression is one or more of these arguments:

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>numeric</td>
<td>A numeric value. Numeric values can include decimals and negative values.</td>
</tr>
<tr>
<td>row, column, or</td>
<td>A pointer to a row, column, or cell within a grid. References can be specified in several ways. The reference syntax is: GridName.GridElement[segment(range)].Property.</td>
</tr>
<tr>
<td>reference</td>
<td></td>
</tr>
<tr>
<td>function</td>
<td>An embedded function.</td>
</tr>
<tr>
<td>operators</td>
<td>Use any of the supported arithmetic operators (+, -, *, /, ^, %, and so on).</td>
</tr>
</tbody>
</table>

**Example**

The following example divides row 1 by row 2 and then rounds the data to 4 places:

\[
\text{Round(Eval([1]/[2]),4)}
\]
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 is as follows:

\[
\text{IfThen} (\text{Condition}, \text{TrueParameter}, \text{FalseParameter})
\]

- **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>
<thead>
<tr>
<th>Conditional Operator</th>
<th>Syntax</th>
<th>Logic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equal To</td>
<td>expression = expression</td>
<td>Tests if the left expression is equal to the right expression.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note</strong>: The routine that evaluates the condition does not consider any rounding. If rounding is required, use the Round function.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Example</strong>:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = 4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Returns false</td>
</tr>
<tr>
<td>Greater Than</td>
<td>expression &gt; expression</td>
<td>Tests if the left expression is greater than the right expression.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Example</strong>:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 &gt; 4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Returns false</td>
</tr>
<tr>
<td>Greater Than or Equal To</td>
<td>expression &gt;= expression</td>
<td>Tests if the left expression is greater than or equal to the right expression.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note</strong>: The correct syntax is &quot;(\geq)&quot;. The syntax &quot;(\geq)&quot; is not supported.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Example</strong>:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 &gt;= 4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Returns false</td>
</tr>
<tr>
<td>Less Than</td>
<td>expression &lt; expression</td>
<td>Tests if the left expression is less than the right expression.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Example</strong>:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 &lt; 4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Returns true</td>
</tr>
<tr>
<td>Conditional Operator</td>
<td>Syntax</td>
<td>Logic</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Less Than or Equal To</td>
<td>expression &lt;= expression</td>
<td>Tests if the left expression is less than or equal to the right expression.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> The correct syntax is &quot;(&lt;=)&quot;. The syntax &quot;(=&lt;=)&quot; is not supported.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 &lt;= 4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Returns true</td>
</tr>
<tr>
<td>Not Equal To</td>
<td>expression &lt;&gt; expression</td>
<td>Tests if the left expression is not equal to the right expression.</td>
</tr>
<tr>
<td></td>
<td>expression != expression</td>
<td><strong>Note:</strong> The routine that evaluates the condition does not consider any rounding. If rounding is required, use the Round function.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 &lt;&gt; 4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Returns true</td>
</tr>
<tr>
<td></td>
<td>IsMiss (reference)</td>
<td><strong>Note:</strong> If the reference is an expanded row or column, then all resulting cells must be #MISSING in order for the condition to be true.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>IsMissing([1])</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Returns true if row 1 has a #MISSING value.</td>
</tr>
<tr>
<td>IsError</td>
<td>IsError (reference)</td>
<td>Tests if the reference contains an #ERROR result.</td>
</tr>
<tr>
<td></td>
<td>IsErr (reference)</td>
<td><strong>Note:</strong> If the reference is an expanded row or column, all resulting cells must be #ERROR in order for the condition to be true.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>IsError([2])</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Returns true if row 2 has a #ERROR value.</td>
</tr>
<tr>
<td>IsNonNumeric</td>
<td>IsNN (reference)</td>
<td>Tests if the reference contains a #MISSING or #ERROR results.</td>
</tr>
<tr>
<td></td>
<td>IsNonNumber (reference)</td>
<td><strong>Note:</strong> 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.</td>
</tr>
<tr>
<td></td>
<td>IfNN (reference)</td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>IfNonNumber (reference)</td>
<td>IsNN([3])</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Returns true if row 3 has a #MISSING or #ERROR value.</td>
</tr>
<tr>
<td>Parenthesis</td>
<td>(condition)</td>
<td>Used to group a condition. Mostly used for visual clarity.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(1 &gt; 4)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Returns false</td>
</tr>
</tbody>
</table>
### Table 96  Conditional Operators

<table>
<thead>
<tr>
<th>Complex Conditions</th>
<th>Syntax</th>
<th>Logic</th>
</tr>
</thead>
</table>
| And                | (condition AND condition) | Complex condition used to compare two conditions. Returns true if all conditions result in true.  
**Example:**  
(1 > 4 AND 5 > 2)  
Returns false |
| Not                | NOT (condition)          | Used to negate the result by reversing the result of the condition.  
**Example:**  
Not (1 > 4)  
Returns true |
| Or                 | (condition OR condition) | Complex condition used to compare two conditions. Returns true if any of the conditions result in true.  
**Example:**  
(1 > 4 OR 5 > 2)  
Returns true |

### Notes on Conditions

- **Expression** can be any valid formula expression. The expression can be any combination of a constant (integer or real number), a reference, or another function.
- **Reference** can be any valid reference; thus the IFNN reference property can be utilized as part of the reference.
- **Condition** can be 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 is:

Max (arguments)

where arguments is one or more of these arguments:

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>numeric</td>
<td>A numeric value. For example, Max (10, 20, 30) returns the value 30. Numeric values can include decimals and negative values.</td>
</tr>
<tr>
<td>row, column, or cell reference</td>
<td>A pointer to a row, column, or cell within a grid. References can be specified in several ways. The reference syntax is: GridName.GridElement[segment(range)].Property.</td>
</tr>
<tr>
<td>function</td>
<td>An embedded function.</td>
</tr>
</tbody>
</table>

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 is as follows:

Min (arguments)

where arguments is one or more of these arguments:

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>numeric</td>
<td>A numeric value. For example, Min (10, 20, 30) returns the value 10. Numeric values can include decimals and negative values.</td>
</tr>
<tr>
<td>row, column, or cell reference</td>
<td>A pointer to a row, column, or cell within a grid. References can be specified in several ways. The reference syntax is: GridName.GridElement[segment(range)].Property</td>
</tr>
<tr>
<td>function</td>
<td>An embedded function.</td>
</tr>
</tbody>
</table>

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:
**Mod**

Mod is a mathematical function that returns the remainder, or modulus, from a division. The function syntax is:

\[ \text{Mod} (\text{arg1}, \text{arg2}) \]

where \text{arg2} is the divisor and \text{arg1} and \text{arg2} are one of these arguments:

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>numeric</td>
<td>A numeric value. For example, Mod (6, 4) returns the value 2. Numeric values can include decimals and negative values.</td>
</tr>
<tr>
<td>row, column, or cell reference</td>
<td>A pointer to a row, column, or cell within a grid. References can be specified in several ways. The reference syntax is: GridName.GridElement[segment(range)].Property</td>
</tr>
<tr>
<td>function</td>
<td>An embedded function.</td>
</tr>
</tbody>
</table>

Example:

The following example divides 10 by 5 and returns the remainder of 0:

\[ \text{Mod} (10, 5) = 0 \]

**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 is:

\[ \text{PercentOfTotal} (\text{arg1}, \text{arg2}) \]

- where \text{arg1} is a component of the running total (\text{arg2}). Normally, this is a row or column reference.
- where \text{arg2} is the running total relative to \text{arg1}. Normally this is a cell reference containing the grand total.
- \text{arg1} is divided by \text{arg2}, with the result multiplied by 100. \text{arg1} and \text{arg2} are one or more of these arguments:

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>numeric</td>
<td>A numeric value. For example, PercentOfTotal (100, 20) returns the value 500. Numeric values can include decimals and negative values.</td>
</tr>
<tr>
<td>row, column, or cell reference</td>
<td>A pointer to a row, column, or cell within a grid. References can be specified in several ways. The reference syntax is: GridName.GridElement[segment(range)].Property</td>
</tr>
<tr>
<td>function</td>
<td>An embedded function.</td>
</tr>
</tbody>
</table>
**Note:** This function requires two arguments.

**Examples:**
The following example returns the value of 5 percent.

\[
\text{PercentOfTotal}(20, 400)
\]

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:

\[
\text{PercentOfTotal}([A], [A,5]),
\]

Using the above example, the following table shows the PercentOfTotal results in column B:

<table>
<thead>
<tr>
<th></th>
<th><strong>A</strong></th>
<th><strong>B</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Sales</td>
<td>% Total</td>
</tr>
<tr>
<td>2</td>
<td>Mkt1</td>
<td>60</td>
</tr>
<tr>
<td>3</td>
<td>Mkt2</td>
<td>120</td>
</tr>
<tr>
<td>4</td>
<td>Mkt3</td>
<td>120</td>
</tr>
<tr>
<td>5</td>
<td>Total Mkt</td>
<td>300</td>
</tr>
</tbody>
</table>

**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 is:

\[
\text{PI()}
\]

**Example:**
The following example returns row 3 divided by the product of Pi and 2:

\[
\text{row[3]} / (\text{PI()} * 2)
\]

**Product**

Product is a mathematical function that multiplies all numbers or references and returns the product. The function syntax is:

\[
\text{Product}(\text{arguments})
\]

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 is: `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 is:

```
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 is as follows:

**Rank([Reference], Order)**

**Rank([Reference], Order, Unique)**

<table>
<thead>
<tr>
<th><strong>Argument</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Reference</strong></td>
<td>The range of cells, rows, or columns to rank, with letters identifying columns and numbers identifying rows. For example, specify <code>[A:1:5]</code> to rank the values for rows 1 through 5 in column A. You can use the <code>.ifNN</code> 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 <code>.ifNN(-1)</code> to assign the value -1 to any cell with a missing value.</td>
</tr>
</tbody>
</table>
### Argument Description

**Order**
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:
- Ascending
- Descending
- Asc
- Des
- Desc
- 1 (the number 1 is the same as "ascending")
- 0 (zero is the same as "descending")

The keywords are not case-sensitive.

**Note:** Do not enclose the number or keyword indicating order in quotation marks.

**Unique (optional)**
Optional. A Boolean keyword indicating how to treat equal values in the Reference parameter where:
- 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.

### 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:

<table>
<thead>
<tr>
<th>East</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cola</td>
<td>16</td>
</tr>
<tr>
<td>Fruit Drinks</td>
<td>23</td>
</tr>
<tr>
<td>Beer</td>
<td>16</td>
</tr>
<tr>
<td>Diet</td>
<td>missing</td>
</tr>
<tr>
<td>Root Beer</td>
<td>0</td>
</tr>
</tbody>
</table>

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:
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):

<table>
<thead>
<tr>
<th></th>
<th>East</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cola</td>
<td>16</td>
<td>2</td>
</tr>
<tr>
<td>Fruit Drinks</td>
<td>23</td>
<td>1</td>
</tr>
<tr>
<td>Beer</td>
<td>16</td>
<td>2</td>
</tr>
<tr>
<td>Diet</td>
<td>missing</td>
<td>5</td>
</tr>
<tr>
<td>Root Beer</td>
<td>0</td>
<td>4</td>
</tr>
</tbody>
</table>

**Round**

Round is a mathematical function that rounds a number up or down by the specified digits. The function syntax is:

```
Round (arg1, integer)
```

where `arg1` is one or more of these arguments:

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>numeric</td>
<td>A numeric value. For example, Round(81.3987,3) returns the value 81.399. Numeric values can include decimals and negative values.</td>
</tr>
<tr>
<td>row, column, or cell</td>
<td>A pointer to a row, column, or cell within a grid. References can be specified in several ways. The reference syntax is: GridName.GridElement[segment(range)].property</td>
</tr>
</tbody>
</table>

438 Form Formula Functions
**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 3 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 is:

```
Sqrt (argument)
```

where `argument` is one of these arguments:

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>numeric</td>
<td>A numeric value. For example, Sqrt(100) returns the value 10. Numeric values can include decimals and negative values.</td>
</tr>
<tr>
<td>row, column, or cell reference</td>
<td>A pointer to a row, column, or cell within a grid. References can be specified in several ways. The reference syntax is: GridName.GridElement[segment(range)].property</td>
</tr>
<tr>
<td>function</td>
<td>An embedded function.</td>
</tr>
</tbody>
</table>

**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 is:

**Sum** (arguments)

where **arguments** is one or more of these arguments:

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>numeric</td>
<td>A numeric value. For example, Sum(10, 20, 30) returns the value 60. Numeric values can include decimals and negative values.</td>
</tr>
<tr>
<td>row, column, or cell reference</td>
<td>A pointer to a row, column, or cell within a grid. References can be specified in several ways. The reference syntax is: GridName.GridElement[segment(range)].property</td>
</tr>
<tr>
<td>function</td>
<td>An embedded function.</td>
</tr>
</tbody>
</table>

Examples:

- The following example returns the value 30:
  
  \[ \text{sum}(10, 20) \]

- The following example returns the sum of three rows:
  
  \[ \text{sum}(	ext{row}[1], \text{row}[6], \text{row}[8]) \]

- The following example calculates the sum of three aggregate columns:
  
  \[ \text{sum}(	ext{column}[E], \text{column}[G], \text{column}[I]) \]

When a formula row includes \text{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):

\text{IDESC(“Mem1”), IDESC(“Mem2”), IDESC(“Mem3”), IDESC(“Mem4”)}

If a formula row is added with the following formula:

\[ \text{SUM(ROW[1],ROW[2],ROW[3],ROW[4])} \]

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:**

\[ \text{Trunc} (arg1, \text{integer}) \]

- where **arg1** is one of these arguments:

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>numeric</td>
<td>A numeric value. For example, 234.567.</td>
</tr>
</tbody>
</table>
**Argument** | **Description**
--- | ---
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 is: `GridName.GridElement[segment(range)].property`
function | 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 2 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>
<thead>
<tr>
<th>Column A</th>
<th>Column B</th>
<th>( \text{Var} ([A], [B]) = 0 )</th>
<th>( \text{Var} ([A], [B]) &gt; 0 )</th>
<th>( \text{Var} ([A], [B]) &lt; 0 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asset</td>
<td>Asset</td>
<td>0</td>
<td>Returns a positive value</td>
<td>Returns a negative value</td>
</tr>
<tr>
<td>Liability</td>
<td>Liability</td>
<td>0</td>
<td>Returns a positive value</td>
<td>Returns a negative value</td>
</tr>
<tr>
<td>Equity</td>
<td>Equity</td>
<td>0</td>
<td>Returns a positive value</td>
<td>Returns a negative value</td>
</tr>
<tr>
<td>Revenue</td>
<td>Revenue</td>
<td>0</td>
<td>Returns a positive value</td>
<td>Returns a negative value</td>
</tr>
<tr>
<td>Expense</td>
<td>Expense</td>
<td>0</td>
<td>Returns a negative value</td>
<td>Returns a positive value</td>
</tr>
</tbody>
</table>

**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:

<table>
<thead>
<tr>
<th>Sales</th>
<th>Expense</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>-400</td>
<td>100</td>
<td>-500</td>
</tr>
</tbody>
</table>

- 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.

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample syntax referencing a column:</td>
<td>( \text{Var} ([A], [B]) )</td>
</tr>
<tr>
<td>Sample syntax referencing a row:</td>
<td>( \text{Var} ([3], [4]) )</td>
</tr>
<tr>
<td>Sample syntax referencing a cell:</td>
<td>( \text{Var} (\text{Cell} [3,A], [3,B]) )</td>
</tr>
</tbody>
</table>

In this example, the variance between column A (Actual) and column B (Budget) is calculated as:

\( \text{Var} ([A], [B]) \)

This example produces the following report:
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:

\[ \text{VarPer} (\text{reference1, reference2}) \]

where \text{reference1} and \text{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>
<thead>
<tr>
<th>Col A</th>
<th>Col B</th>
<th>VarPer ([A] , [B])=0</th>
<th>VarPer ([A] , [B])&gt;0</th>
<th>VarPer ([A] , [B])&lt;0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asset</td>
<td>Asset</td>
<td>0</td>
<td>Returns a positive value</td>
<td>Returns a negative value</td>
</tr>
<tr>
<td>Liability</td>
<td>Liability</td>
<td>0</td>
<td>Returns a negative value</td>
<td>Returns a positive value</td>
</tr>
<tr>
<td>Equity</td>
<td>Equity</td>
<td>0</td>
<td>Returns a positive value</td>
<td>Returns a negative value</td>
</tr>
<tr>
<td>Revenue</td>
<td>Revenue</td>
<td>0</td>
<td>Returns a positive value</td>
<td>Returns a negative value</td>
</tr>
<tr>
<td>Expense</td>
<td>Expense</td>
<td>0</td>
<td>Returns a negative value</td>
<td>Returns a positive value</td>
</tr>
</tbody>
</table>

VariancePercent Behavior

- The VariancePercent function expects comparison of the same account type. When you compare two different account types, like Sales & Expense, the VariancePercent function performs the straight math without applying the logic of the account type. For example:

<table>
<thead>
<tr>
<th>Sales</th>
<th>Expense</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>-400</td>
<td>100</td>
<td>-5.</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sample syntax referencing a column:</strong></td>
<td>VarPer ([A], [B])</td>
</tr>
<tr>
<td><strong>Sample syntax referencing a row:</strong></td>
<td>VarPer ([3], [4])</td>
</tr>
<tr>
<td><strong>Sample syntax referencing a cell:</strong></td>
<td>VarPer (Cell [3,A], [3,B])</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Year</th>
<th>Product</th>
<th>Market</th>
<th>Actual</th>
<th>Budget</th>
<th>VariancePercent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sales (Income)</td>
<td>400,855</td>
<td>373,080</td>
<td>7%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>COGS (Expense)</td>
<td>179,336</td>
<td>158,940</td>
<td>-13%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
In This Appendix

- Importing Planning Dimensions in Smart View .............................................. 445
- Editing Members in Smart View ................................................................. 447
- Adding Planning Members in Smart View .................................................... 448
- Moving Members in Smart View ................................................................. 450
- Guidelines for Moving Members in Smart View .......................................... 451
- Designating Shared Members in Smart View .............................................. 451
- Refreshing and Creating Cubes in Smart View ........................................... 452

Note: All procedures described in this appendix are performed within the Smart View application.

You can use the Smart View application to quickly perform several tasks with your Planning metadata:

- “Importing Planning Dimensions in Smart View” on page 445
- “Editing Members in Smart View” on page 447
- “Adding Planning Members in Smart View” on page 448
- “Moving Members in Smart View” on page 450
- “Designating Shared Members in Smart View” on page 451
- “Refreshing and Creating Cubes in Smart View” on page 452

For additional information on accessing Smart View and understanding Smart View functionality and commands, see Oracle Smart View for Office User's Guide.

**Importing Planning Dimensions in Smart View**

Importing a Planning dimension into Smart View 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” on page 446
Importing Dimensions in Smart View

Note: You must have administrator privileges to import dimensions in Smart View.

To import a 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. Double-click a dimension name or right-click the dimension name and select Edit Dimension. A Smart View grid listing the dimension members is displayed.

Using Smart View Grids

The Smart View grid allows you to rapidly add, edit, and move the members of a Planning dimension.

Overview of the Smart View Grid

A Smart View grid consists of two dimensions:

- A 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 dimension 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 Smart View grid displays the 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” on page 447.
The ribbon displayed here is similar to the ribbon in a conventional (non-Planning) Smart View grid but contains fewer options. The enabled buttons include:

- Zoom In
- Zoom Out
- Keep Only
- Remove Only
- Member Selection
- Refresh
- Submit Data

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 Smart View grid to edit 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.
- Once a metadata grid is opened, it cannot be re-linked with 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 Smart View grid allows you to rapidly edit the properties of members of a 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” on page 446).
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 Smart View grid allows you to rapidly add members to a 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” on page 448
- “Adding Members in Smart View” on page 449
- “Guidelines for Adding Members in Smart View” on page 450

Selecting the Add Mode within Planning Application Properties

There are two different modes for adding new members in the Smart View 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 or star (*) 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 or star (*) in the grid after you perform the Refresh. You can use a different sign by adding SMART_VIEW_DIMENSION_EDITOR_NEW_MEMBER_SUFFIX in 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” on page 48.)
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 Smart View grid allows you to rapidly add members to a 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” on page 446).
2 Enter the name of a new member in the name column.
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 or star (*) 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” on page 447.)

**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:** To modify the properties of the new member, see “Editing Members in Smart View” on page 447.

**Note:** 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 of no significance.
- 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 Planning to a new member. The specific default properties are based upon those of the dimension.

### Moving Members in Smart View

The Smart View 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 Planning dimension into a Smart View grid (see “Importing Dimensions in Smart View” on page 446.)
2 Highlight the member in the **Parent Member** column in the grid.

**Note:** An empty Parent Member property will cause 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 sub trees in the same **Submit Data** operation. If the Submit Data operation fails for one member, the Planning server will stop the operation and not save any 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 Smart View.

---

**Designating Shared Members in Smart View**

The Smart View 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 Smart View grid allows you to quickly refresh a cube or create a new one.

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 Oracle Smart View for Office 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.
Glossary

! See bang character.

#MISSING See missing data.

access permissions A set of operations that a user can perform on a resource.

account blocking The process by which accounts accept input data in the consolidated file. Blocked accounts do not receive their value through the additive consolidation process.

account eliminations Accounts which have their values set to zero in the consolidated file during consolidation.

account type A property that determines how an account’s value flows over time and its sign behavior. Account type options can include expense, income, asset, liability, and equity.

accountability map A visual, hierarchical representation of the responsibility, reporting, and dependency structure of the accountability teams (also known as critical business areas) in an organization.

active service A service whose Run Type is set to Start rather than to Hold.

active-active high availability system A system in which all the available members can service requests, and no member is idle. An active-active system generally provides more scalability options than an active-passive system. Contrast with active-passive high availability system.

active-passive high availability system A system with active members, which are always servicing requests, and passive members that are activated only when an active member fails. Contrast with active-active high availability system.

activity-level authorization Defines user access to applications and the types of activities they can perform on applications, independent of the data that will be operated on.

ad hoc report An online analytical query that an end user creates dynamically.

adapter Software that enables a program to integrate with data and metadata from target and source systems.

adaptive states Interactive Reporting Web Client level of permission.

adjustment See journal entry.

Advanced Relational Access The integration of a relational database with an Essbase multidimensional database so that all data remains in the relational database and is mapped to summary-level data in the Essbase database.

agent An Essbase server process that starts and stops applications and databases, manages connections from users, and handles user-access security. The agent is referred to as ESSBASE.EXE.

aggregate cell A cell comprising several cells. For example, a data cell that uses Children(Year) expands to four cells containing Quarter 1, Quarter 2, Quarter 3, and Quarter 4 data.

aggregate function A type of function, such as sum or calculation of an average, that summarizes or performs analysis on data.

aggregate limit A limit placed on an aggregated request line item or aggregated metatopic item.

aggregate storage database The database storage model designed to support large-scale, sparsely distributed data which is categorized into many, potentially large dimensions. Upper level members and formulas are dynamically calculated, and selected data values are aggregated and stored, typically with improvements in overall aggregation time.
aggregate view - A collection of aggregate cells based on the levels of the members within each dimension. To reduce calculation time, values are pre-aggregated and stored as aggregate views. Retrievals start from aggregate view totals and add up from there.

aggregation - The process of rolling up and storing values in an aggregate storage database; the stored result of the aggregation process.

aggregation script - In aggregate storage databases only, a file that defines a selection of aggregate views to be built into an aggregation.

alias table - A table that contains alternate names for members.

alternate hierarchy - A hierarchy of shared members. An alternate hierarchy is based upon an existing hierarchy in a database outline, but has alternate levels in the dimension. An alternate hierarchy allows the same data to be seen from different points of view.

ancestor - A branch member that has members below it. For example, the members Qtr2 and 2006 are ancestors of the member April.

appender - A Log4j term for destination.

application - 1) A software program designed to run a specific task or group of tasks such as a spreadsheet program or database management system; 2) A related set of dimensions and dimension members that are used to meet a specific set of analytical requirements, reporting requirements, or both.

application administrator - A person responsible for setting up, configuring, maintaining, and controlling an application. Has all application privileges and data access permissions.

application currency - The default reporting currency for the application.

Application Migration Utility - A command-line utility for migrating applications and artifacts.

application server cluster - A loosely joined group of application servers running simultaneously, working together for reliability and scalability, and appearing to users as one application server instance. See also vertical application cluster and horizontal application cluster.

area - A predefined set of members and values that makes up a partition.

arithmetic data load - A data load that performs operations on values in the database, such as adding 10 to each value.

artifact - An individual application or repository item; for example, scripts, forms, rules files, Interactive Reporting documents, and financial reports. Also known as an object.

assemblies - Installation files for EPM System products or components.

asset account - An account type that stores values that represent a company’s assets.

assignment - The association of a source and destination in the allocation model that controls the direction of allocated costs or revenue flow.

asymmetric topology - An Oracle Fusion Middleware Disaster Recovery configuration that is different across tiers on the production site and standby site. For example, an asymmetric topology can include a standby site with fewer hosts and instances than the production site.

attribute - A characteristic of a dimension member. For example, Employee dimension members may have attributes of Name, Age, or Address. Product dimension members can have several attributes, such as a size and flavor.

attribute association - A relationship in a database outline whereby a member in an attribute dimension describes a characteristic of a member of its base dimension. For example, if product 100-10 has a grape flavor, the product 100-10 has the Flavor attribute association of grape. Thus, the 100-10 member of the Product dimension is associated with the Grape member of the Flavor attribute dimension.

Attribute Calculations dimension - A system-defined dimension that performs these calculation operations on groups of members: Sum, Count, Avg, Min, and Max. This dimension is calculated dynamically and is not visible in the database outline. For example, using the Avg member, you can calculate the average sales value for Red products in New York in January.

attribute dimension - A type of dimension that enables analysis based on the attributes or qualities of dimension members.

attribute reporting - A reporting process based on the attributes of the base dimension members. See also base dimension.
attribute type  A text, numeric, Boolean, date, or linked-
attribute type that enables different functions for grouping,
selecting, or calculating data. For example, because the
Ounces attribute dimension has the type numeric, the
number of ounces specified as the attribute of each product
can be used to calculate the profit per ounce for that
product.
authentication  Verification of identity as a security measure.
Authentication is typically based on a user name and
password. Passwords and digital signatures are forms of
authentication.
authentication service  A core service that manages one
authentication system.
auto-reversing journal  A journal for entering adjustments that
you want to reverse in the next period.
automated stage  A stage that does not require human
intervention; for example, a data load.
axis  1) A straight line that passes through a graphic used for
measurement and categorization; 2) A report aspect used to
arrange and relate multidimensional data, such as filters,
pages, rows, and columns. For example, for a data query in
Simple Basic, an axis can define columns for values for Qtr1,
Qtr2, Qtr3, and Qtr4. Row data would be retrieved with
totals in the following hierarchy: Market, Product.
backup  A duplicate copy of an application instance.
balance account  An account type that stores unsigned values
that relate to a particular time.
balanced journal  A journal in which the total debits equal the
total credits.
bang character (!)  A character that terminates a series of
report commands and requests information from the
database. A report script must be terminated with a bang
character; several bang characters can be used within a
report script.
base currency  The currency in which daily business
transactions are performed.
base dimension  A standard dimension that is associated with
one or more attribute dimensions. For example, assuming
products have flavors, the Product dimension is the base
dimension for the Flavors attribute dimension.
base entity  An entity at the bottom of the organization
structure that does not own other entities.
batch calculation  Any calculation on a database that is done
in batch; for example, a calculation script or a full database
calculation. Dynamic calculations are not considered to be
batch calculations.
batch file  An operating system file that can call multiple
ESSCMD scripts and run multiple sessions of ESSCMD. On
Windows-based systems, batch files have BAT file
extensions. On UNIX, batch files are written as a shell script.
Batch Loader  An FDM component that enables the
processing of multiple files.
batch POV  A collection of all dimensions on the user POV of
every report and book in the batch. While scheduling the
batch, you can set the members selected on the batch POV.
batch processing mode  A method of using ESSCMD to write
a batch or script file that can be used to automate routine
server maintenance and diagnostic tasks. ESSCMD script
files can execute multiple commands and can be run from
the operating system command line or from within
operating system batch files. Batch files can be used to call
multiple ESSCMD scripts or run multiple instances of
ESSCMD.
block  The primary storage unit which is a multidimensional
array representing the cells of all dense dimensions.
block storage database  The Essbase database storage model
categorizing and storing data based on the sparsity of data
values defined in sparse dimensions. Data values are stored
in blocks, which exist only for sparse dimension members
for which there are values.
Blocked Account  An account that you do not want calculated
in the consolidated file because you want to enter it
manually.
book  1) In Financial Reporting, a container that holds a
group of similar documents. Books may specify dimension
sections or dimension changes; 2) In Data Relationship
Management, a collection of exports that can be run
together as a group. Export results can be combined
together or output separately.
book POV  The dimension members for which a book is run.
**bookmark** A link to a reporting document or a Web site, displayed on a personal page of a user. The types of bookmarks are My Bookmarks and image bookmarks.

**bounding rectangle** The required perimeter that encapsulates the Interactive Reporting document content when embedding Interactive Reporting document sections in a personal page, specified in pixels for height and width or row per page.

**broadcast message** A simple text message sent by an administrator to a user who is logged on to a Planning application. The message details information such as system availability, notification of application refresh, or application backups.

**build method** A method used to modify database outlines. Choice of a build method is based on the format of data in data source files.

**business process** A set of activities that collectively accomplish a business objective.

**business rules** Logical expressions or formulas that are created within an application to produce a desired set of resulting values.

**cache** A buffer in memory that holds data temporarily.

**calc script** A set of commands that define how a database is consolidated or aggregated. A calculation script may also contain commands that specify allocation and other calculation rules separate from the consolidation process.

**Calculated Accounts** Accounts with formulas that you cannot alter. These formulas are fixed to maintain the accounting integrity of the model that you are building. For example, the formula for Net Income, a Calculated Account, is modeled into Strategic Finance and cannot be changed in historical or forecast periods.

**calculated member in MaxL DML** A member designed for analytical purposes and defined in the optional WITH section of a MaxL DML query.

**Calculation Manager** A module of Enterprise Performance Management Architecture (EPMA) that Planning and Financial Management users can use to design, validate, and administrate business rules in a graphical environment.

**calculation status** A consolidation status that indicates that some values or formula calculations have changed. You must reconsolidate to get the correct values for the affected entity.

**calendar** User-defined time periods and their relationship to each other. Q1, Q2, Q3, and Q4 comprise a calendar or fiscal year.

**cascade** The process of creating multiple reports for a subset of member values.

**Catalog pane** An area that displays a list of elements available to the active section. If Query is the active section, a list of database tables is displayed. If Pivot is the active section, a list of results columns is displayed. If Dashboard is the active section, a list of embeddable sections, graphic tools, and control tools are displayed.

**categories** Groupings by which data is organized. For example, Month.

**cause and effect map** A map that depicts how the elements that form your corporate strategy relate and how they work together to meet your organization’s strategic goals. A Cause and Effect map tab is automatically created for each Strategy map.

**CDF** See custom-defined function.

**CDM** See custom-defined macro.

**cell** 1) The data value at the intersection of dimensions in a multidimensional database; the intersection of a row and a column in a worksheet; 2) A logical group of nodes belonging to one administrative domain.

**cell note** A text annotation for a cell in an Essbase database. Cell notes are a type of LRO.

**CHANGED status** Consolidation status that indicates data for an entity has changed.

**chart template** A template that defines the metrics to display in Workspace charts.

**child** A member with a parent above it in the database outline.
choice list  A list of members that a report designer can specify for each dimension when defining the report’s point of view. A user who wants to change the point of view for a dimension that uses a choice list can select only the members specified in that defined member list or those members that meet the criteria defined in the function for the dynamic list.

clean block  A data block in which the database is fully calculated, if a calculation script calculates all dimensions at once, or if the SET CLEARUPDATESTATUS command is used in a calculation script.

cluster  An array of servers or databases that behave as a single resource which share task loads and provide failover support; eliminates one server or database as a single point of failure in a system.

cluster interconnect  A private link used by a hardware cluster for heartbeat information, to detect node failure.

cluster services  Software that manages cluster member operations as a system. With cluster services, you can define a set of resources and services to monitor through a heartbeat mechanism between cluster members and to move these resources and services to a different cluster member as efficiently and transparently as possible.

clustered bar charts  Charts in which categories are viewed side-by-side; used only with vertical bar charts.

code page  A mapping of bit combinations to a set of text characters. Different code pages support different sets of characters. Each computer contains a code page setting for the character set requirements of the language of the computer user. In the context of this document, code pages map characters to bit combinations for non-Unicode encodings. See also encoding.

column  In Data Relationship Management, a field of data associated with an import source or the results of a query, compare, validation, or export.

committed access  An Essbase Kernel Isolation Level setting that affects how Essbase handles transactions. Under committed access, concurrent transactions hold long-term write locks and yield predictable results.

computed item  A virtual column (as opposed to a column that is physically stored in the database or cube) that can be calculated by the database during a query, or by Interactive Reporting Studio in the Results section. Computed items are calculations of data based on functions, data items, and operators provided in the dialog box and can be included in reports or reused to calculate other data.

connection file  See Interactive Reporting connection file (.oce)

consolidated file (Parent)  A file into which all of the business unit files are consolidated; contains the definition of the consolidation.

consolidation  The process of aggregating data from dependent entities to parent entities. For example, if the dimension Year consists of the members Qtr1, Qtr2, Qtr3, and Qtr4, its consolidation is Year.

consolidation file (*.cns)  A graphical interface that enables you to add, delete, or move Strategic Finance files in the consolidation process using either a Chart or Tree view. It also enables you to define and modify the consolidation.

consolidation rule  The rule that is executed during the consolidation of the node of the hierarchy. This rule can contain customer-specific formulas appropriate for the correct consolidation of parent balances. Elimination processing can be controlled within these rules.

content  Information stored in the repository for any type of file.

content browser  A component that enables users to browse and select content to be placed on a Workspace Page.

context variable  A variable that is defined for a particular task flow to identify the context of the taskflow instance.

contribution  The value added to a parent from a child entity. Each child has a contribution to its parent.

controls groups  Groupings used in FDM to maintain and organize certification and assessment information, especially helpful for meeting Sarbanes-Oxley requirements.

conversion rate  See exchange rate.

cookie  A segment of data placed on your computer by a Web site.
correlated subqueries  Subqueries that are evaluated once for every row in the parent query; created by joining a topic item in the subquery with a topic in the parent query.

critical business area (CBA)  An individual or a group organized into a division, region, plant, cost center, profit center, project team, or process; also called accountability team or business area.

critical success factor (CSF)  A capability that must be established and sustained to achieve a strategic objective; owned by a strategic objective or a critical process and is a parent to one or more actions.

crosstab reporting  Reporting that categorizes and summarizes data in table format. The table cells contain summaries of the data that fit within the intersecting categories. For example, a crosstab report of product sales information could show size attributes, such as Small and Large, as column headings and color attributes, such as Blue and Yellow, as row headings. The cell in the table where Large and Blue intersect could contain the total sales of all Blue products that are sized Large.

cube  A block of data that contains three or more dimensions. An Essbase database is a cube.

cube deployment  In Essbase Studio, the process of setting load options for a model to build an outline and load data into an Essbase application and database.

cube schema  In Essbase Studio, the metadata elements, such as measures and hierarchies, representing the logical model of a cube.

currency conversion  A process that converts currency values in a database from one currency into another. For example, to convert one U. S. dollar into the European euro, the exchange rate (for example, 0.923702) is multiplied by the dollar (1 * 0.923702). After conversion, the European euro amount is 0.92.

Currency Overrides  A feature allowing the selected input method for any input period to be overridden to enable input of that period’s value as Default Currency/Items. To override the input method, enter a pound sign (#) before or after the number.

currency partition  A dimension type that separates local currency members from a base currency, as defined in an application. Identifies currency types, such as Actual, Budget, and Forecast.

custom calendar  Any calendar created by an administrator.

custom dimension  A dimension created and defined by users. Channel, product, department, project, or region could be custom dimensions.

custom property  A property of a dimension or dimension member that is created by a user.

custom report  A complex report from the Design Report module, composed of any combination of components.

custom-defined function (CDF)  Essbase calculation functions developed in Java and added to the standard Essbase calculation scripting language using MaxL. See also custom-defined macro.

custom-defined macro (CDM)  Essbase macros written with Essbase calculator functions and special macro functions. Custom-defined macros use an internal Essbase macro language that enables the combination of calculation functions and they operate on multiple input parameters. See also custom-defined function.

cycle through  Perform multiple passes through a database while calculating it.

dashboard  A collection of metrics and indicators that provide an interactive summary of your business. Dashboards enable you to build and deploy analytic applications.

data cache  A buffer in memory that holds uncompressed data blocks.

data cell  See cell.

data file cache  A buffer in memory that holds compressed data (PAG) files.

data function  Function that computes aggregate values, including averages, maximums, counts, and other statistics that summarize groupings of data.

data load location  In FDM, a reporting unit responsible for submitting source data into the target system. Typically, one FDM data load location exists for each source file loaded to the target system.

data load rules  A set of criteria that determines how to load data from a text-based file, a spreadsheet, or a relational data set into a database.
**data lock** A feature that prevents changes to data according to specified criteria, such as a period or scenario.

**data model** A representation of a subset of database tables.

**data value** See cell.

**database connection** A file that stores definitions and properties used to connect to data sources and enables database references to be portable and widely used.

**date measure** In Essbase, a member tagged as Date in the dimension where measures are represented. The cell values are displayed as formatted dates. Dates as measures can be useful for analysis types that are difficult to represent using the Time dimension. For example, an application may need to track acquisition dates for a series of capital assets, but the acquisition dates span too large a period to allow for feasible Time dimension modeling. See also typed measure.

**Default Currency Units** The unit scale of data. For example, if you select to define your analysis in thousands and enter 10, this unit is interpreted as 10,000.

**dense dimension** In block storage databases, a dimension likely to contain data for every combination of dimension members. For example, time dimensions are often dense because they can contain all combinations of all members. Contrast with sparse dimension.

**dependent entity** An entity that is owned by another entity in the organization.

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

**descendant** Any member below a parent in the database outline. In a dimension that includes years, quarters, and months, the members Qtr2 and April are descendants of the member Year.

**Design Report** An interface in Web Analysis Studio for designing custom reports, from a library of components.

**destination** 1) In Business Rules, a block of the database where calculated values are stored; 2) In Profitability and Cost Management, the association of a source and destination in the allocation model that controls the direction of allocated costs or revenue flow.

**destination currency** The currency to which balances are converted. You enter exchange rates and convert from the source currency to the destination currency. For example, when you convert from EUR to USD, the destination currency is USD.

**detail chart** A chart that provides the detailed information that you see in a Summary chart. Detail charts appear in the Investigate Section in columns below the Summary charts. If the Summary chart shows a Pie chart, then the Detail charts below represent each piece of the pie.

**dimension** A data category used to organize business data for the retrieval and preservation of values. Dimensions usually contain hierarchies of related members grouped within them. For example, a Year dimension often includes members for each time period, such as quarters and months.

**dimension build** The process of adding dimensions and members to an Essbase outline.

**dimension build rules** Specifications, similar to data load rules, that Essbase uses to modify an outline. The modification is based on data in an external data source file.

**dimension tab** In the Pivot section, the tab that enables you to pivot data between rows and columns.

**dimension table** 1) A table that includes numerous attributes about a specific business process; 2) In Essbase Integration Services, a container in the OLAP model for one or more relational tables that define a potential dimension in Essbase.

**dimension type** A dimension property that enables the use of predefined functionality. Dimensions tagged as time have a predefined calendar functionality.

**dimensionality** In MaxL DML, the represented dimensions (and the order in which they are represented) in a set. For example, the following set consists of two tuples of the same dimensionality, because they both reflect the dimensions (Region, Year): \{ (West, Feb), (East, Mar) \}
**direct rate**  A currency rate that you enter in the exchange-rate table. The direct rate is used for currency conversion. For example, to convert balances from JPY to USD, in the exchange-rate table, enter a rate for the period/scenario where the source currency is JPY and the destination currency is USD.

**dirty block**  A data block containing cells that have been changed since the last calculation. Upper-level blocks are marked as dirty if their child blocks are dirty (that is, if they have been updated).

**Disaster Recovery**  The ability to safeguard against natural or unplanned outages at a production site by having a recovery strategy for applications and data to a geographically separate standby site.

**display type**  One of three Web Analysis formats saved to the repository: spreadsheet, chart, and pinboard.

**dog-ear**  The flipped page corner in the upper-right corner of the chart header area.

**drill-down**  Navigation through the query result set using the dimensional hierarchy. Drilling down moves the user perspective from aggregated data to detail. For example, drilling down can reveal hierarchical relationships between years and quarters or quarters and months.

**drill-through**  The navigation from a value in one data source to corresponding data in another source.

**driver**  In Profitability and Cost Management, an allocation method that describes the mathematical relationship between the sources that use the driver and the destinations to which those sources allocate cost or revenue. For Business Modeling, see also cost driver and activity driver.

**duplicate alias name**  A name that occurs more than once in an alias table and can be associated with more than one member in a database outline. Duplicate alias names can be used with duplicate member outlines only.

**duplicate member name**  Multiple occurrences of a member name in a database, with each occurrence representing a different member. For example, a database has two members named New York. One member represents New York state and the other member represents New York city.

**duplicate member outline**  A database outline containing duplicate member names.

**Dynamic Calc and Store members**  Members in a block storage outline that Essbase calculates only upon the first retrieval of the value. Essbase then stores the calculated value in the database. Subsequent retrievals do not require calculating.

**Dynamic Calc members**  Members in a block storage outline that Essbase calculates only at retrieval time. Essbase discards calculated values after completing the retrieval request.

**dynamic calculation**  In Essbase, a calculation that occurs only when you retrieve data on a member that is tagged as Dynamic Calc or Dynamic Calc and Store. The member’s values are calculated at retrieval time instead of being precalculated during batch calculation.

**dynamic hierarchy**  In aggregate storage database outlines only, a hierarchy in which members are calculated at retrieval time.

**dynamic member list**  A system-created named member set that is based on user-defined criteria. The list is refreshed automatically whenever it is referenced in the application. As dimension members are added and deleted, the list automatically reapplies the criteria to reflect the changes.

**dynamic reference**  A pointer in the rules file to header records in a data source.

**dynamic report**  A report containing data that is updated when you run the report.

**Dynamic Time Series**  A process that performs period-to-date reporting in block storage databases.

**dynamic view account**  An account type indicating that account values are calculated dynamically from the data that is displayed.

**Eliminated Account**  An account that does not appear in the consolidated file.

**elimination**  The process of zeroing out (eliminating) transactions between entities within an organization.

**employee**  A user responsible for, or associated with, specific business objects. Employees need not work for an organization; for example, they can be consultants. Employees must be associated with user accounts, for authorization purposes.
encoding  A method for mapping bit combinations to characters for creating, storing, and displaying text. Each encoding has a name; for example, UTF-8. Within an encoding, each character maps to a specific bit combination; for example, in UTF-8, uppercase A maps to HEX41. See also code page, locale.

ending period  A period enabling you to adjust the date range in a chart. For example, an ending period of "month" produces a chart showing information through the end of the current month.

Enterprise View  An Administration Services feature that enables management of the Essbase environment from a graphical tree view. From Enterprise View, you can operate directly on Essbase artifacts.

text  A dimension representing organizational units. Examples: divisions, subsidiaries, plants, regions, products, or other financial reporting units.

EPM Oracle home  A subdirectory of Middleware home containing the files required by EPM System products. The EPM Oracle home location is specified during installation with EPM System Installer.

EPM Oracle instance  A directory containing active, dynamic components of EPM System products (components that can change during runtime). You define the EPM Oracle instance directory location during configuration with EPM System Configurator.

Equity Beta  The riskiness of a stock, measured by the variance between its return and the market return, indicated by an index called "beta." For example, if a stock's return normally moves up or down 1.2% when the market moves up or down 1%, the stock has a beta of 1.2.

essbase.cfg  An optional configuration file for Essbase. Administrators may edit this file to customize Essbase Server functionality. Some configuration settings may also be used with Essbase clients to override Essbase Server settings.

EssCell  A function entered into a cell in Essbase Spreadsheet Add-in to retrieve a value representing an intersection of specific Essbase database members.

ESSCMD  A command-line interface for performing Essbase operations interactively or through batch script files.

ESSLANG  The Essbase environment variable that defines the encoding used to interpret text characters. See also encoding.

ESSMSH  See MaxL Shell.

exceptions  Values that satisfy predefined conditions. You can define formatting indicators or notify subscribing users when exceptions are generated.

exchange rate type  An identifier for an exchange rate. Different rate types are used because there may be multiple rates for a period and year. Users traditionally define rates at period end for the average rate of the period and for the end of the period. Additional rate types are historical rates, budget rates, forecast rates, and so on. A rate type applies to a specific time.

expense account  An account that stores periodic and year-to-date values that decrease net worth if they are positive.

Extensible Markup Language (XML)  A language comprising a set of tags used to assign attributes to data that can be interpreted between applications according to a schema.

external authentication  Logging on to Oracle EPM System products with user information stored outside the application. The user account is maintained by the EPM System, but password administration and user authentication are performed by an external service, using a corporate directory such as Oracle Internet Directory (OID) or Microsoft Active Directory (MSAD).

externally triggered events  Non-time-based events for scheduling job runs.

Extract, Transform, and Load (ETL)  Data-source-specific programs for extracting data and migrating it to applications.

extraction command  An Essbase reporting command that handles the selection, orientation, grouping, and ordering of raw data extracted from a database; begins with the less-than (<) character.

fact table  The central table in a star join schema, characterized by a foreign key and elements drawn from a dimension table. This table typically contains numeric data that can be related to all other tables in the schema.
failover  The ability to switch automatically to a redundant standby database, server, or network if the primary database, server, or network fails or is shut down. A system that is clustered for failover provides high availability and fault tolerance through server redundancy and fault-tolerant hardware, such as shared disks.

Favorites gadget  A gadget that contains links to Reporting and Analysis documents and URLs. See also gadget.

file delimiter  A character, such as a comma or tab, that separates fields in a data source.

filter  A constraint on data sets that restricts values to specific criteria; for example, to exclude certain tables, metadata, or values, or to control access.

flow account  An unsigned account that stores periodic and year-to-date values.

footer  Text or images at the bottom of report pages, containing dynamic functions or static text such as page numbers, dates, logos, titles or file names, and author names.

form  A grid display that enables users to enter data into the database from an interface such as a Web browser, and to view and analyze data or related text. Certain dimension member values are fixed, giving users a specific view into the data.

format string  1) In Essbase, a method for transforming the way cell values are displayed; 2) In Data Relationship Management, a parameter of a Format or Formatted Date derived property that indicates the format in which a property value should be returned.

formula  In Data Relationship Management, business logic used by a derived property to dynamically calculate a property value.

frame  An area on the desktop. Two main areas: the navigation and workspace frames.

free-form grid  An object for presenting, entering, and integrating data from different sources for dynamic calculations.

free-form reporting  Creating reports by entering dimension members or report script commands in worksheets.

function  In Data Relationship Management, a syntactic element of a derived property formula that accepts parameters and returns dynamic values.

gadget  A simple, specialized, lightweight application that provides easy viewing of EPM content and enables access to core Reporting and Analysis functionality.

genealogy data  Additional data that is optionally generated after allocation calculations. This data enables reporting on all cost or revenue flows from start to finish through all allocation steps.

generation  A layer in a hierarchical tree structure that defines member relationships in a database. Generations are ordered incrementally from the top member of the dimension (generation 1) down to the child members. Use the unique generation name to identify a layer in the hierarchical tree structure.

generic jobs  Non-SQR Production Reporting or non-Interactive Reporting jobs.

global report command  A command in a running report script that is effective until it is replaced by another global command or the file ends.

grid POV  A means for specifying dimension members on a grid without placing dimensions in rows, columns, or page intersections. A report designer can set POV values at the grid level, preventing user POVs from affecting the grid. If a dimension has one grid value, you put the dimension into the grid POV instead of the row, column, or page.

group  A container for assigning similar access permissions to multiple users.

GUI  Graphical user interface

hardware cluster  A collection of computers that provides a single view of network services (for example, an IP address) or application services (such as databases and Web servers) to clients of these services. Each node in a hardware cluster is a standalone server that runs its own processes. These processes can communicate with one another to form what looks like a single system that cooperatively provides applications, system resources, and data to users.
**high availability** A system attribute that enables an application to continue to provide services in the presence of failures. This is achieved through removal of single points of failure, with fault-tolerant hardware, as well as server clusters; if one server fails, processing requests are routed to another server.

**Historical Average** An average for an account over a number of historical periods.

**holding company** An entity that is part of a legal entity group, with direct or indirect investments in all entities in the group.

**horizontal application server cluster** A cluster with application server instances on different machines.

**host** A server on which applications and services are installed.

**host properties** Properties pertaining to a host, or if the host has multiple Oracle EPM homes, to an Oracle EPM home.

**Hybrid Analysis** An analysis mapping low-level data stored in a relational database to summary-level data stored in Essbase, combining the mass scalability of relational systems with multidimensional data.

**hyperlink** A link to a file, a Web page, or an intranet HTML page.

**Hypertext Markup Language (HTML)** A programming language specifying how Web browsers display data.

**identity** A unique identification for a user or group in external authentication.

**image bookmarks** Graphic links to Web pages or repository items.

**IMPACTED status** A status that indicates changes in child entities consolidating into parent entities.

**implied share** A member with one or more children but only one that is consolidated, so the parent and child share a value.

**import format** In FDM, the definition of the structure of the source file that enables the loading of a source data file to an FDM data-load location.

**inactive group** A group for which an administrator has deactivated system access.

**INACTIVE status** A status that indicates entities deactivated from consolidation for the current period.

**inactive user** A user whose account was deactivated by an administrator.

**income account** An account storing periodic and year-to-date values that, if positive, increase net worth.

**index** 1) A method where Essbase uses sparse-data combinations to retrieve data in block storage databases. 2) The index file.

**index cache** A buffer containing index pages.

**index entry** A pointer to an intersection of sparse dimensions. Index entries point to data blocks on disk and use offsets to locate cells.

**index file** An Essbase file storing block storage data retrieval information, residing on disk, and containing index pages.

**index page** A subdivision in an index file. An index page contains pointers to data blocks.

**input data** Data loaded from a source rather than calculated.

**installation assemblies** Product installation files that plug into EPM System Installer.

**integration** A process that is run to move data between Oracle’s Hyperion applications using Shared Services. Data integration definitions specify the data moving between a source application and a destination application, and they enable the data movements to be grouped, ordered, and scheduled.

**intelligent calculation** A calculation method tracking updated data blocks since the last calculation.

**Interactive Reporting connection file (.oce)** Files encapsulating database connection information, including the database API (ODBC, SQL*Net, and so on), database software, the database server network address, and database user name. Administrators create and publish Interactive Reporting connection (.oce) files.

**intercompany elimination** See elimination.
intercompany matching  The process of comparing balances for pairs of intercompany accounts within an application. Intercompany receivables are compared to intercompany payables for matches. Matching accounts are used to eliminate intercompany transactions from an organization’s consolidated totals.

intercompany matching report  A report that compares intercompany account balances and indicates whether the accounts are in balance.

interdimensional irrelevance  A situation in which a dimension does not intersect with other dimensions. Because the data in the dimension cannot be accessed from the nonintersecting dimensions, the nonintersecting dimensions are not relevant to that dimension.

intersection  A unit of data representing the intersection of dimensions in a multidimensional database; also, a worksheet cell.

intrasstage assignment  An assignment in the financial flow to an object within the same stage.

introspection  A deep inspection of a data source to discover hierarchies based on the inherent relationships in the database. Contrast with scraping.

Investigation  See drill-through.

isolation level  An Essbase Kernel setting that determines the lock and commit behavior of database operations. Choices are: committed access and uncommitted access.

iteration  A pass of the budget or planning cycle in which the same version of data is revised and promoted.

Java application server cluster  An active-active application server cluster of Java Virtual Machines (JVMs).

Java Database Connectivity (JDBC)  A client-server communication protocol used by Java-based clients and relational databases. The JDBC interface provides a call-level API for SQL-based database access.

job output  Files or reports produced from running a job.

jobs  Documents with special properties that can be launched to generate output. A job can contain Interactive Reporting, SQR Production Reporting, or generic documents.

Join  A link between two relational database tables or topics based on common content in a column or row. A join typically occurs between identical or similar items within different tables or topics. For example, a record in the Customer table is joined to a record in the Orders table because the Customer ID value is the same in each table.

journal entry (JE)  A set of debit-credit adjustments to account balances for a scenario and period.

JSP  Java Server Page.

KeyContacts gadget  A gadget that contains a group of Smart Space users and provides access to Smart Space Collaborator. For example, you can have a KeyContacts gadget for your marketing team and another for your development team. See also gadget.

latest  A spreadsheet keyword used to extract data values from the member defined as the latest time period.

layer  1) The horizontal location of members in a hierarchical structure, specified by generation (top down) or level (bottom up); 2) Position of objects relative to other objects. For example, in the Sample Basic database, Qtr1 and Qtr4 are in the same layer, so they are also in the same generation, but in a database with a ragged hierarchy, Qtr1 and Qtr4 might not be in the same layer, though they are in the same generation.

layout area  An area on a Workspace Page where content can be placed.

legend box  A box containing labels that identify the data categories of a dimension.

level  A layer in a hierarchical tree structure that defines database member relationships. Levels are ordered from the bottom dimension member (level 0) up to the parent members.

level 0 block  A data block for combinations of sparse, level 0 members.

level 0 member  A member that has no children.

liability account  An account type that stores "point in time" balances of a company's liabilities. Examples: accrued expenses, accounts payable, and long-term debt.

lifecycle management  The process of migrating an application, a repository, or individual artifacts across product environments.
**line item detail**  The lowest level of detail in an account.

**lineage**  The relationship between different metadata elements showing how one metadata element is derived from one or more other metadata elements, ultimately tracing the metadata element to its physical source. In Essbase Studio, a lineage viewer displays the relationships graphically. See also traceability.

**link**  1) A reference to a repository object. Links can reference folders, files, shortcuts, and other links; 2) In a taskflow, the point where the activity in one stage ends and another begins.

**link condition**  A logical expression evaluated by the taskflow engine to determine the sequence of launching taskflow stages.

**linked data model**  Documents that are linked to a master copy in a repository.

**linked partition**  A shared partition that enables you to use a data cell to link two databases. When a user clicks a linked cell in a worksheet, Essbase opens a new sheet displaying the dimensions in the linked database. The user can then drill down those dimensions.

**linked reporting object (LRO)**  A cell-based link to an external file such as cell notes, URLs, or files with text, audio, video, or pictures. (Only cell notes are supported for Essbase LROs in Financial Reporting.) Contrast with local report object.

**load balancer**  Hardware or software that directs the requests to individual application servers in a cluster and is the only point of entry into the system.

**load balancing**  Distribution of requests across a group of servers, which helps to ensure optimal end user performance.

**local currency**  An input currency type. When an input currency type is not specified, the local currency matches the entity’s base currency.

**local report object**  A report object that is not linked to a Financial Reporting report object in Explorer. Contrast with linked reporting object.

**local results**  A data model’s query results. Results can be used in local joins by dragging them into the data model. Local results are displayed in the catalog when requested.

**locale**  A computer setting that specifies a location’s language, currency and date formatting, data sort order, and the character set encoding used on the computer. Essbase uses only the encoding portion. See also encoding, ESSLANG.

**locale header record**  A text record at the beginning of some non-Unicode-encoded text files, such as scripts, that identifies the encoding locale.

**location alias**  A descriptor that identifies a data source. The location alias specifies a server, application, database, user name, and password. Location aliases are set by DBAs at the database level using Administration Services Console, ESSCMD, or the API.

**locked**  A user-invoked process that prevents users and processes from modifying data.

**locked data model**  A data model that cannot be modified by a user.

**LOCKED status**  A consolidation status indicating that an entity contains data that cannot be modified.

**Log Analyzer**  An Administration Services feature that enables filtering, searching, and analysis of Essbase logs.

**logic group**  In FDM, one or more logic accounts generated after a source file is loaded into FDM. Logic accounts are calculated accounts derived from the source data.

**logical Web application**  An aliased reference used to identify the internal host name, port, and context of a Web application. In a clustered or high-availability environment, this is the alias name that establishes a single internal reference for the distributed components. In EPM System, a nonclustered logical Web application defaults to the physical host running the Web application.

**LRO**  See linked reporting object.

**managed server**  An application server process running in its own Java Virtual Machine (JVM).

**manual stage**  A stage that requires human intervention.

**Map File**  A file that stores the definition for sending data to or retrieving data from an external database. Map files have different extensions (.mps to send data; .mpr to retrieve data).
Map Navigator  A feature that displays your current position on a Strategy, Accountability, or Cause and Effect map, indicated by a red outline.

Marginal Tax Rate  The rate used to calculate the after-tax cost of debt; represents the tax rate applied to the last earned income dollar (the rate from the highest tax bracket into which income falls) and includes federal, state, and local taxes. Based on current level of taxable income and tax bracket, you can predict marginal tax rate.

Market Risk Premium  The additional rate of return paid over the risk-free rate to persuade investors to hold "riskier" investments than government securities. Calculated by subtracting the risk-free rate from the expected market return. These figures should closely model future market conditions.

Master Data Model  An independent data model that is referenced as a source by multiple queries. When used, "Locked Data Model" is displayed in the Query section's Content pane; the data model is linked to the master data model displayed in the Data Model section, which an administrator may hide.

Mathematical Operator  A symbol that defines how data is calculated in formulas and outlines. Can be any of the standard mathematical or Boolean operators; for example, +, -, *, /, and %.

MaxL  The multidimensional database access language for Essbase, consisting of a data definition language (MaxL DDL) and a data manipulation language (MaxL DML). See also MaxL DDL, MaxL DML, and MaxL Shell.

MaxL DDL  The data definition language used by Essbase for batch or interactive system-administration tasks.

MaxL DML  The data manipulation language used in Essbase for data query and extraction.

MaxL Perl Module  A Perl module (essbase.pm) that is part of Essbase MaxL DDL. This module can be added to the Perl package to provide access to Essbase databases from Perl programs.

MaxL Script Editor  A script-development environment in Administration Services Console. MaxL Script Editor is an alternative to using a text editor and the MaxL Shell for administering Essbase with MaxL scripts.

MaxL Shell  An interface for passing MaxL statements to Essbase Server. The MaxL Shell executable file is located in the Essbase bin directory (UNIX: essmsh; Windows: essmsh.exe).

MDX (Multidimensional Expression)  A language used for querying and calculation in multidimensional-compliant databases.

Measures  Numeric values in an OLAP database cube that are available for analysis. Measures are margin, cost of goods sold, unit sales, budget amount, and so on. See also fact table.

Member  A discrete component within a dimension. A member identifies and differentiates the organization of similar units. For example, a time dimension might include members Jan, Feb, and Qtr1.

Member List  A named system- or user-defined group that references members, functions, or member lists within a dimension.

Member Load  In Essbase Integration Services, the process of adding dimensions and members (without data) to Essbase outlines.

Member Selection Report Command  A type of Report Writer command that selects member ranges based on outline relationships, such as sibling, generation, and level.

Member-Specific Report Command  A type of Report Writer formatting command that is executed as it is encountered in a report script. The command affects only its associated member and executes the format command before processing the member.

Merge  A data load option that clears values only from the accounts specified in the data load file and replaces them with values in the data load file.

Metadata  A set of data that defines and describes the properties and attributes of the data stored in a database or used by an application. Examples of metadata are dimension names, member names, properties, time periods, and security.

Metadata Elements  Metadata derived from data sources and other metadata that is stored and cataloged for Essbase Studio use.

Metadata Sampling  The process of retrieving a sample of members in a dimension in a drill-down operation.
metadata security  Security set at the member level to restrict
users from accessing certain outline members.

metaoutline  In Essbase Integration Services, a template
containing the structure and rules for Essbase outline from
an OLAP model.

Middleware home  A directory that includes the Oracle
WebLogic Server home and can also include the EPM Oracle
home and other Oracle homes. A Middleware home can
reside on a local file system or on a remote shared disk that
is accessible through NFS.

migration audit report  A report generated from the migration
log that provides tracking information for an application
migration.

migration definition file (.mdf)  A file that contains migration
parameters for an application migration, enabling batch
script processing.

migration log  A log file that captures all application migration
actions and messages.

migration snapshot  A snapshot of an application migration
that is captured in the migration log.

MIME Type  An attribute that describes the data format of an
item, so that the system knows which application should
open the object. A file's MIME (Multipurpose Internet Mail
Extension) type is determined by the file extension or HTTP
header. Plug-ins tell browsers which MIME types they
support and which file extensions correspond to each
MIME type.

minireport  A report component that includes layout,
content, hyperlinks, and the query or queries to load the
report. Each report can include one or more minireports.

minischema  A graphical representation of a subset of tables
from a data source that represents a data modeling context.

missing data (#MISSING)  A marker indicating that data in the
labeled location does not exist, contains no value, or was
never entered or loaded. For example, missing data exists
when an account contains data for a previous or future
period but not for the current period.

model  1) A file or content string containing an application-
specific representation of data. Models are the basic data
managed by Shared Services, of two major types:
dimensional and nondimensional application objects; 2) In
Business Modeling, a network of boxes connected to
represent and calculate the operational and financial flow
through the area being examined.

multidimensional database  A method of organizing, storing,
and referencing data through three or more dimensions. An
individual value is the intersection point for a set of
dimensions. Contrast with relational database.

My Workspace Page  Customizable Workspace Pages created
by users. They are marked specially so that they can be easily
accessed from one single place without having to navigate
the repository.

named set  In MaxL DML, a set with its logic defined in the
optional WITH section of a MaxL DML query. The named
set can be referenced multiple times in the query.

native authentication  The process of authenticating a user
name and password from within the server or application.

nested column headings  A report column heading format that
displays data from multiple dimensions. For example, a
column heading that contains Year and Scenario members
is a nested column. The nested column heading shows Q1
(from the Year dimension) in the top line of the heading,
qualified by Actual and Budget (from the Scenario
dimension) in the bottom line of the heading.

NO DATA status  A consolidation status indicating that this
entity contains no data for the specified period and account.

non-dimensional model  A Shared Services model type that
includes application objects such as security files, member
lists, calculation scripts, and Web forms.

non-unique member name  See duplicate member name.

null value  A value that is absent of data. Null values are not
equal to zero.
numeric attribute range. A feature used to associate a base dimension member that has a discrete numeric value with an attribute that represents a value range. For example, to classify customers by age, an Age Group attribute dimension can contain members for the following age ranges: 0-20, 21-40, 41-60, and 61-80. Each Customer dimension member can be associated with an Age Group range. Data can be retrieved based on the age ranges rather than on individual age values.

ODBC Open Database Connectivity. A database access method used from any application regardless of how the database management system (DBMS) processes the information.

OK status A consolidation status indicating that an entity has already been consolidated, and that data has not changed below it in the organization structure.

OLAP Metadata Catalog In Essbase Integration Services, a relational database containing metadata describing the nature, source, location, and type of data that is pulled from the relational data source.

OLAP model In Essbase Integration Services, a logical model (star schema) that is created from tables and columns in a relational database. The OLAP model is then used to generate the structure of a multidimensional database. See also online analytical processing (OLAP).

online analytical processing (OLAP) A multidimensional, multiuser, client-server computing environment for users who analyze consolidated enterprise data in real time. OLAP systems feature drill-down, data pivoting, complex calculations, trend analysis, and modeling.

Open Database Connectivity (ODBC) Standardized application programming interface (API) technology that allows applications to access multiple third-party databases.

Oracle home A directory containing the installed files required by a specific product, and residing within the directory structure of Middleware home. See also Middleware home.

organization An entity hierarchy that defines each entity and their relationship to others in the hierarchy.

origin The intersection of two axes.

outline The database structure of a multidimensional database, including all dimensions, members, tags, types, consolidations, and mathematical relationships. Data is stored in the database according to the structure defined in the outline.

outline synchronization For partitioned databases, the process of propagating outline changes from one database to another database.

P&L accounts (P&L) Profit and loss accounts. P&L refers to a typical grouping of expense and income accounts that comprise a company's income statement.

page A display of information in a grid or table often represented by the Z-axis. A page can contain data from one field, derived data from a calculation, or text.

page file An Essbase data file.

page heading A report heading type that lists members represented on the current page of the report. All data values on the page have the members in the page heading as a common attribute.

page member A member that determines the page axis.

palette A JASC-compliant file with a .PAL extension. Each palette contains 16 colors that complement each other and can be used to set the dashboard color elements.

parallel calculation A calculation option. Essbase divides a calculation into tasks and calculates some tasks simultaneously.

parallel data load In Essbase, the concurrent execution of data load stages by multiple process threads.

parallel export The ability to export Essbase data to multiple files. This may be faster than exporting to a single file, and it may resolve problems caused by a single data file becoming too large for the operating system to handle.

parent adjustments The journal entries that are posted to a child in relation to its parent.

parents The entities that contain one or more dependent entities that report directly to them. Because parents are entities associated with at least one node, they have entity, node, and parent information associated with them.
partition area  A subcube within a database. A partition is composed of one or more areas of cells from a portion of the database. For replicated and transparent partitions, the number of cells within an area must be the same for the data source and target to ensure that the two partitions have the same shape. If the data source area contains 18 cells, the data target area must also contain 18 cells to accommodate the number of values.

partitioning  The process of defining areas of data that are shared or linked between data models. Partitioning can affect the performance and scalability of Essbase applications.

pattern matching  The ability to match a value with any or all characters of an item entered as a criterion. Missing characters may be represented by wild-card values such as a question mark (?) or an asterisk (*). For example, "Find all instances of apple" returns apple, but "Find all instances of apple*" returns apple, applesauce, applecranberry, and so on.

percent consolidation  The portion of a child’s values that is consolidated to its parent.

percent control  The extent to which an entity is controlled within the context of its group.

percent ownership  The extent to which an entity is owned by its parent.

performance indicator  An image file used to represent measure and scorecard performance based on a range you specify; also called a status symbol. You can use the default performance indicators or create an unlimited number of your own.

periodic value method (PVA)  A process of currency conversion that applies the periodic exchange rate values over time to derive converted results.

permission  A level of access granted to users and groups for managing data or other users and groups.

perspective  A category used to group measures on a scorecard or strategic objectives within an application. A perspective can represent a key stakeholder (such as a customer, employee, or shareholder/financial) or a key competency area (such as time, cost, or quality).

pinboard  One of the three data object display types. Pinboards are graphics composed of backgrounds and interactive icons called pins. Pinboards require traffic lighting definitions.

pins  Interactive icons placed on graphic reports called pinboards. Pins are dynamic. They can change images and traffic lighting color based on the underlying data values and analysis tools criteria.

pivot  Alter the perspective of retrieved data. When Essbase first retrieves a dimension, it expands data into rows. You can then pivot or rearrange the data to obtain a different viewpoint.

planner  A user who can input and submit data, use reports that others create, execute business rules, use task lists, enable email notification for themselves, and use Smart View. Planners comprise the majority of users.

planning unit  A data slice at the intersection of a scenario, version, and entity; the basic unit for preparing, reviewing, annotating, and approving plan data.

plot area  The area bounded by X, Y, and Z axes; for pie charts, the rectangular area surrounding the pie.

plug account  An account in which the system stores any out-of-balance differences between intercompany account pairs during the elimination process.

post stage assignment  Assignments in the allocation model that are assigned to locations in a subsequent model stage.
POV (point of view)  A feature for setting data focus by selecting members that are not already assigned to row, column, or page axes. For example, selectable POVs in FDM could include location, period, category, and target category. In another example, using POV as a filter in Smart View, you could assign the Currency dimension to the POV and select the Euro member. Selecting this POV in forms displays data in Euro values.

precalculation  Calculating the database before user retrieval.

precision  Number of decimal places displayed in numbers.

predefined drill paths  Paths used to drill to the next level of detail, as defined in the data model.

presentation  A playlist of Web Analysis documents, enabling reports to be grouped, organized, ordered, distributed, and reviewed. Includes pointers referencing reports in the repository.

preserve formulas  User-created formulas kept within a worksheet while retrieving data.

primary measure  A high-priority measure important to your company and business needs. Displayed in the Contents frame.

Process Monitor Report  A list of locations and their positions within the FDM data conversion process. You can use the process monitor report to monitor the status of the closing process. The report is time-stamped. Therefore, it can be used to determine to which locations at which time data was loaded.

product  In Shared Services, an application type, such as Planning or Performance Scorecard.

Production Reporting  See SQR Production Reporting.

project  An instance of Oracle’s Hyperion products grouped together in an implementation. For example, a Planning project may consist of a Planning application, an Essbase cube, and a Financial Reporting Server instance.

provisioning  The process of granting users and groups specific access permissions to resources.

proxy server  A server acting as an intermediary between workstation users and the Internet to ensure security.

public job parameters  Reusable named job parameters created by administrators and accessible through the access control system.

public recurring time events  Reusable time events created by administrators and accessible through the access control system.

PVA  See periodic value method.

qualified name  A member name in a qualified format that differentiates duplicate member names in a duplicate member outline. For example, [Market].[East].[State]. [New York] or [Market].[East].[City].[New York].

query governor  An Essbase Integration Server parameter or Essbase Server configuration setting that controls the duration and size of queries made to data sources.

reciprocal assignment  An assignment in the financial flow that also has the source as one of its destinations.

reconfigure URL  A URL that is used to reload servlet configuration settings dynamically when users are already logged on to the Workspace.

record  In a database, a group of fields making up one complete entry. For example, a customer record may contain fields for name, address, telephone number, and sales data.

recurring template  A journal template for making identical adjustments in every period.

recurring time event  An event specifying a starting point and the frequency for running a job.

redundant data  Duplicate data blocks that Essbase retains during transactions until Essbase commits updated blocks.

regular journal  A feature for entering one-time adjustments for a period. A regular journal can be balanced, balanced by entity, or unbalanced.

Related Accounts  Accounts related to the main account and grouped under the same main account number. The account structure groups all main and related accounts under the same main account number. The main account is distinguished from related accounts by the first suffix of the account number.

relational database  A type of database that stores data in related two-dimensional tables. Contrast with multidimensional database.
replace  A data load option that clears existing values from all accounts for periods specified in the data load file and loads values from the data load file. If an account is not specified in the load file, its values for the specified periods are cleared.

replicated partition  A portion of a database, defined through Partition Manager, used to propagate an update to data mastered at one site to a copy of data stored at another site. Users can access the data as though it were part of their local database.

Report Extractor  An Essbase component that retrieves report data from the Essbase database when report scripts are run.

report object  In report designs, a basic element with properties defining behavior or appearance, such as text boxes, grids, images, and charts.

report script  A text file containing Essbase Report Writer commands that generate one or more production reports.

Report Viewer  An Essbase component that displays complete reports after report scripts are run.

reporting currency  The currency used to prepare financial statements, and converted from local currencies to reporting currencies.

repository  Storage location for metadata, formatting, and annotation information for views and queries.

resources  Objects or services managed by the system, such as roles, users, groups, files, and jobs.

restore  An operation to reload data and structural information after a database has been damaged or destroyed, typically performed after shutting down and restarting the database.

restructure  An operation to regenerate or rebuild the database index and, in some cases, data files.

result frequency  The algorithm used to create a set of dates to collect and display results.

review level  A Process Management review status indicator representing the process unit level, such as Not Started, First Pass, Submitted, Approved, and Published.

Risk Free Rate  The rate of return expected from "safer" investments such as long-term U.S. government securities.

role  The means by which access permissions are granted to users and groups for resources.

roll-up  See consolidation.

root member  The highest member in a dimension branch.

runtime prompt  A variable that users enter or select before a business rule is run.

sampling  The process of selecting a representative portion of an entity to determine the entity’s characteristics. See also metadata sampling.

saved assumptions  User-defined Planning assumptions that drive key business calculations (for example, the cost per square foot of office floor space).

scaling  Scaling determines the display of values in whole numbers, tens, hundreds, thousands, millions, and so on.

scenario  A dimension for classifying data; for example, Actuals, Budget, Forecast1, or Forecast2.

schema  In relational databases, a logical model that represents the data and the relationships between the data.

scope  The area of data encompassed by any Essbase operation or setting; for example, the area of data affected by a security setting. Most commonly, scope refers to three levels of granularity, where higher levels encompass lower levels. The levels, from highest to lowest: the entire system (Essbase Server), applications on Essbase Server, or databases within Essbase Server applications. See also persistence.

score  The level at which targets are achieved, usually expressed as a percentage of the target.

scorecard  A business object that represents the progress of an employee, strategy element, or accountability element toward goals. Scorecards ascertain this progress based on data collected for each measure and child scorecard added to the scorecard.

scraping  An inspection of a data source to derive the most basic metadata elements from it. Contrast with introspection.

secondary measure  A low-priority measure, less important than primary measures. Secondary measures do not have Performance reports but can be used on scorecards and to create dimension measure templates.
security agent A Web access management provider (for example, Oracle Access Manager, Oracle Single Sign-On, or CA SiteMinder) that protects corporate Web resources.

security platform A framework enabling Oracle EPM System products to use external authentication and single sign-on.

serial calculation The default calculation setting. Divides a calculation pass into tasks and calculates one task at a time.

services Resources that enable business items to be retrieved, changed, added, or deleted. Examples: Authorization and Authentication.

servlet A piece of compiled code executable by a Web server.

shared disks See shared storage.

shared member A member that shares storage space with another member of the same name, preventing duplicate calculation of members that occur multiple times in an Essbase outline.

Shared Services Registry The part of the Shared Services repository that manages EPM System deployment information for most EPM System products, including installation directories, database settings, computer names, ports, servers, URLs, and dependent service data.

shared storage A set of disks containing data that must be available to all nodes of a failover cluster; also called shared disks.

Shared Workspace Pages Workspace Pages shared across an organization that are stored in a special System folder and can be accessed by authorized users from the Shared Workspace Pages Navigate menu.

sibling A child member at the same generation as another child member and having the same immediate parent. For example, the members Florida and New York are children of East and each other’s siblings.

silent response files Files providing data that an installation administrator would otherwise be required to provide. Response files enable EPM System Installer or EPM System Configurator to run without user intervention or input.

single point of failure Any component in a system that, if it fails, prevents users from accessing the normal functionality.

single sign-on (SSO) The ability to log on once and then access multiple applications without being prompted again for authentication.

smart tags Keywords in Microsoft Office applications that are associated with predefined actions available from the Smart Tag menu. In Oracle EPM System products, smart tags can also be used to import Reporting and Analysis content and to access Financial Management and Essbase functions.

SmartCut A link to a repository item, in URL form.

snapshot Read-only data from a specific time.

source currency The currency from which values originate and are converted through exchange rates to the destination currency.

sparse dimension In block storage databases, a dimension unlikely to contain data for all member combinations when compared to other dimensions. Contrast with dense dimension. For example, not all customers have data for all products.

SPF files Printer-independent files created by an SQR Production Reporting server, containing a representation of the actual formatted report output, including fonts, spacing, headers, footers, and so on.

Spotlighter A tool that enables color coding based on selected conditions.

SQL spreadsheet A data object that displays the result set of a SQL query.

SQR Production Reporting A specialized programming language for data access, data manipulation, and creating SQR Production Reporting documents.

stage 1) A task description that forms one logical step within a taskflow, usually performed by an individual. A stage can be manual or automated; 2) For Profitability, logical divisions within the model that represent the steps in the allocation process within your organization.

stage action For automated stages, the invoked action that executes the stage.

staging area A database that you create to meet the needs of a specific application. A staging area is a snapshot or restructured version of one or more RDBMS.
**staging table** A database that you create to meet the needs of a specific application. A staging area is a snapshot or restructured version of one or more RDBMSs.

**standard dimension** A dimension that is not an attribute dimension.

**standard journal template** A journal function used to post adjustments that have common adjustment information for each period. For example, you can create a standard template that contains the common account IDs, entity IDs, or amounts, and then use the template as the basis for many regular journals.

**Status bar** The bar at the bottom of the screen that displays helpful information about commands, accounts, and the current status of your data file.

**stored hierarchy** In aggregate storage databases outlines only, a hierarchy in which the members are aggregated according to the outline structure. Stored hierarchy members have certain restrictions; for example, they cannot contain formulas.

**strategic objective (SO)** A long-term goal defined by measurable results. Each strategic objective is associated with one perspective in the application, has one parent, the entity, and is a parent to critical success factors or other strategic objectives.

**Strategy map** Represents how the organization implements high-level mission and vision statements into lower-level, constituent strategic goals and objectives.

**structure view** Displays a topic as a simple list of component data items.

**Structured Query Language** A language used to process instructions to relational databases.

**Subaccount Numbering** A system for numbering subaccounts using nonsequential whole numbers.

**subscribe** Flags an item or folder to receive automatic notification whenever the item or folder is updated.

**Summary chart** In the Investigates Section, a chart that rolls up detail charts shown below in the same column, plotting metrics at the summary level at the top of each chart column.

**supervisor** A user with full access to all applications, databases, related files, and security mechanisms for a server.

**supporting detail** Calculations and assumptions from which the values of cells are derived.

**suppress rows** A setting that excludes rows containing missing values and underscores characters from spreadsheet reports.

**symmetric multiprocessing (SMP)** A server architecture that enables multiprocessing and multithreading. Performance is not significantly degraded when a large number of users simultaneously connect to a single instance.

**symmetric topology** An Oracle Fusion Middleware Disaster Recovery configuration that is identical across tiers on the production site and standby site. In a symmetric topology, the production site and standby site have the identical number of hosts, load balancers, instances, and applications. The same ports are used for both sites. The systems are configured identically and the applications access the same data.

**sync** Synchronization of Shared Services and application models.

**synchronized** The condition that exists when the latest version of a model resides in both the application and in Shared Services. See also model.

**system extract** A feature that transfers data from application metadata into an ASCII file.

**tabs** Navigable views of accounts and reports in Strategic Finance.

**target** Expected results of a measure for a specified period of time (day, quarter, and so on).

**task list** A detailed status list of tasks for a particular user.

**taskflow** The automation of a business process in which tasks are passed from one taskflow participant to another according to procedural rules.

**taskflow definition** Business processes in the taskflow management system that consist of a network of stages and their relationships; criteria indicating the start and end of the taskflow; and information about individual stages, such as participants, associated applications, associated activities, and so on.

**taskflow instance** A single instance of a taskflow including its state and associated data.
**taskflow management system**  A system that defines, creates, and manages the execution of a taskflow, including definitions, user or application interactions, and application executables.

**taskflow participant**  The resource that performs the task associated with the taskflow stage instance for both manual and automated stages.

**Taxes - Initial Balances**  Strategic Finance assumes that the Initial Loss Balance, Initial Gain Balance, and Initial Balance of Taxes Paid entries have taken place in the period before the first Strategic Finance time period.


**text list**  In Essbase, an object that stores text values mapped to numeric identifiers. Text lists enable the use of text measures.

**text measure**  In Essbase, a member tagged as Text in the dimension where measures are represented. The cell values are displayed as predefined text. For example, the text measure Satisfaction Index may have the values Low, Medium, and High. See also typed measure, text list, derived text measure.

**time dimension**  The time period that the data represents, such as fiscal or calendar periods.

**time events**  Triggers for job execution.

**time scale**  A scale that displays metrics by a specific time span, such as monthly or quarterly.

**time series reporting**  A process for reporting data based on a calendar date (for example, year, quarter, month, or week).

**Timeline Viewer**  An FDM feature that enables users to view dates and times of completed process flow steps for specific locations.

**Title bar**  A bar that displays the Strategic Finance name, the file name, and the scenario name Version box.

**toast message**  A message that fades in the lower-right corner of the screen.

**token**  An encrypted identification of one valid user or group on an external authentication system.

**top and side labels**  Column and row headings on the top and sides of a Pivot report.

**top-level member**  A dimension member at the top of the tree in a dimension outline hierarchy, or the first member of the dimension in sort order if there is no hierarchical relationship among dimension members. If a hierarchical relationship exists, the top-level member name is generally the same as the dimension name.

**trace allocations**  A Profitability feature that enables you to visually follow the flow of financial data, either forwards or backwards, from a single intersection throughout the model.

**trace level**  The level of detail captured in a log file.

**traceability**  The ability to track a metadata element to its physical source. For example, in Essbase Studio, a cube schema can be traced from its hierarchies and measure hierarchies to its dimension elements, date/time elements, measures, and, ultimately, to its physical source elements. See also lineage.

**traffic lighting**  Color-coding of report cells, or pins based on a comparison of two dimension members, or on fixed limits.

**transformation**  A process that transforms artifacts so that they function properly in the destination environment after application migration.

**translation**  See currency conversion.

**Transmission Control Protocol/Internet Protocol (TCP/IP)**  A standard set of communication protocols linking computers with different operating systems and internal architectures. TCP/IP utilities are used to exchange files, send mail, and store data to various computers that are connected to local and wide area networks.

**transparent login**  A process that logs in authenticated users without launching the login screen.

**transparent partition**  A shared partition that enables users to access and change data in a remote database as though it is part of a local database.

**triangulation**  A means of converting balances from one currency to another through a third common currency. For example, to convert balances from the Danish krone to the British pound, balances could be converted from the krone to the euro and from the euro to the pound.

**triggers**  An Essbase feature whereby data is monitored according to user-specified criteria that, when met, cause Essbase to alert the user or system administrator.
trusted user  Authenticated user.

tuple  MDX syntax element that references a cell as an intersection of a member from each dimension. If a dimension is omitted, its top member is implied. Examples: (Jan); (Jan, Sales); ([Jan], [Sales], [Cola], [Texas], [Actual]).

two-pass  An Essbase property that is used to recalculate members that are dependent on the calculated values of other members. Two-pass members are calculated during a second pass through the outline.

typed measure  In Essbase, a member tagged as Text or Date in the dimension where measures are represented. The cell values are displayed as predefined text or dates.

unary operator  A mathematical indicator (+, -, *, /, %) associated with an outline member. The unary operator defines how the member is calculated during a database roll-up.

Unicode-mode application  An Essbase application wherein character text is encoded in UTF-8, enabling users with computers set up for different languages to share application data.

unique member name  A nonshared member name that exists only once in a database outline.

unique member outline  A database outline that is not enabled for duplicate member names.

upgrade  The process of deploying a new software release and moving applications, data, and provisioning information from an earlier deployment to the new deployment.

upper-level block  A type of data block wherein at least one of the sparse members is a parent-level member.

user directory  A centralized location for user and group information, also known as a repository or provider. Popular user directories include Oracle Internet Directory (OID), Microsoft Active Directory (MSAD), and Sun Java System Directory Server.

user variable  A variable that dynamically renders forms based on a user's member selection, displaying only the specified entity. For example, a user variable named Department displays specific departments and employees.

user-defined attribute (UDA)  An attribute, associated with members of an outline to describe a characteristic of the members, that can be used to return lists of members that have the specified associated UDA.

user-defined member list  A named, static set of members within a dimension defined by the user.

validation  The process of checking a business rule, report script, or partition definition against the outline to ensure that the object being checked is valid.

validation rules  Rules used in FDM to enforce data integrity. For example, in FDM, validation rules ensure that certain conditions are met after data is loaded from FDM to the target application.

value dimension  A dimension that is used to define input value, translated value, and consolidation detail.

variance  The difference between two values (for example, between planned and actual values).

version  A possible outcome used within the context of a scenario of data. For example, Budget - Best Case and Budget - Worst Case where Budget is scenario and Best Case and Worst Case are versions.

vertical application server cluster  A cluster with multiple application server instances on the same machine.

view  A year-to-date or periodic display of data.

visual cue  A formatted style, such as a font or a color, that highlights specific data value types. Data values may be dimension members; parent, child, or shared members; dynamic calculations; members containing a formula; read-only data cells; read-and-write data cells; or linked objects.

WebLogic Server home  A subdirectory of Middleware home containing installed files required by a WebLogic Server instance. WebLogic Server home is a peer of Oracle homes.

weight  A value assigned to an item on a scorecard that indicates the relative importance of that item in the calculation of the overall scorecard score. The weighting of all items on a scorecard accumulates to 100%. For example, to recognize the importance of developing new features for a product, the measure for New Features Coded on a developer's scorecard would be assigned a higher weighting than a measure for Number of Minor Defect Fixes.
**wild card**  Character that represents any single character (?) or group of characters (*) in a search string.

**WITH section**  In MaxL DML, an optional section of the query used for creating reusable logic to define sets or members. Sets or custom members can be defined once in the WITH section and then referenced multiple times during a query.

**workbook**  An entire spreadsheet file with many worksheets.

**workflow**  The steps required to process data from start to finish in FDM. The workflow consists of Import (loading data from the GL file), Validate (ensures that all members are mapped to a valid account), Export (loads the mapped members to the target application), and Check (verifies accuracy of data by processing data with user-defined validation rules).

**Workspace Page**  A page created with content from multiple sources including documents, URL, and other content types. Enables a user to aggregate content from Oracle and non-Oracle sources.

**write-back**  The ability for a retrieval client, such as a spreadsheet, to update a database value.

**ws.conf**  A configuration file for Windows platforms.

**wsconf_platform**  A configuration file for UNIX platforms.

**XML**  See Extensible Markup Language.

**XOLAP**  An Essbase multidimensional database that stores only the outline metadata and retrieves all data from a relational database at query time. XOLAP supports aggregate storage databases and applications that contain duplicate member names.

**Y axis scale**  A range of values on Y axis of charts displayed in Investigate Section. For example, use a unique Y axis scale for each chart, the same Y axis scale for all Detail charts, or the same Y axis scale for all charts in the column. Often, using a common Y axis improves your ability to compare charts at a glance.

**Zero Administration**  A software tool that identifies version number of the most up-to-date plug-in on the server.

**ZoomChart**  A tool for viewing detailed information by enlarging a chart. A ZoomChart enables you to see detailed numeric information on the metric that is displayed in the chart.