Oracle® Hyperion Planning, Fusion Edition

Administrator's Guide

RELEASE 11.1.2.1

ORACLE®

ENTERPRISE PERFORMANCE
MANAGEMENT SYSTEM

Planning Administrator's Guide, 11.1.2.1

Authors: EPM Information Development Team

Contents

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at http://www.oracle.com/accessibility/.

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Access to Oracle Support for Hearing-Impaired Customers

Oracle customers have access to electronic support through My Oracle Support or by calling Oracle Support at 1.800.223.1711. Hearing-impaired customers in the U.S. who wish to speak to an Oracle Support representative may use a telecommunications relay service (TRS). Information about the TRS is available at http://www.fcc.gov/cgb/consumerfacts/trs.html/, and a list of telephone numbers is available at http://www.fcc.gov/cgb/dro/trsphonebk.html. International hearing-impaired customers should use the TRS at +1.605.224.1837. An Oracle Support engineer will respond to technical issues according to the standard service request process.

1 Using Planning

In This Chapter

About Planning

Oracle Hyperion Planning, Fusion Edition is a Web-based 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:

- Facilitates collaboration, communication, and control across multi-divisional 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 Hyperion Smart View for Office, Fusion Edition
- Lowers the total cost of ownership through a shorter roll out and implementation phase, and easier maintenance for applications
- Enhances decision-making with reporting, analysis, and planning
- Promotes modeling with complex business rules and allocations

- Integrates with other systems to load data

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 Hyperion 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, Fusion Edition.

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 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 Hyperion Enterprise Performance Management System Installation and Configuration Guide*.

EPM Workspace

Planning is available within Oracle Enterprise Performance Management Workspace, Fusion Edition. For information on EPM Workspace features, such as EPM Workspace preferences, see the *Oracle Enterprise Performance Management Workspace User's Online Help* or *Oracle Enterprise*

Performance Management Workspace Administrator's Online Help. To log on to EPM Workspace, see "Logging On to EPM Workspace and Accessing Planning" on page 42.

Lifecycle Management

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

Lifecycle Management features:

- Viewing applications and folders
- Searching for artifacts
- 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 Oracle Hyperion Enterprise Performance Management System Lifecycle Management, see the *Oracle Hyperion 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 Classic Planning applications, see "Managing Data Sources" on page 250. 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, Fusion Edition 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

Business rules perform complicated calculations through predefined formulas, variables, calculation scripts, macros and sequences (for Oracle's Hyperion® Business Rules) and rulesets and templates (for Calculation Manager). See "Using Business Rules" on page 140.

Web Analysis

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

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 21

- “Setting Up Applications” on page 22
- “Building Plans” on page 26
- “Launching Business Rules” on page 27
- “Starting the Budget Review Process” on page 27
- “Post-Planning Activities” on page 29

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 1 Server Installation

User	Task
IT professional	Installs Planning system requirements, including operating systems, relational database, Web server, application server, and Web browser
IT professional	Installs Essbase
IT professional	Sets up the data source name (DSN) for the OLAP and relational databases
IT professional	Installs Planning, Financial Reporting, and (for Classic Planning applications) Business Rules
IT professional	Creates relational databases and connections
IT professional	Configures the Web server
IT professional	Sets up users in an external user directory
IT professional	Optional: Creates a test application with users
IT professional	Tests installation and configuration

Table 2 Client Installation Users and Tasks

User	Task
IT professional	Installs the Planning remote Windows client, Smart View, Microsoft Excel, Financial Reporting, and (for Classic Planning applications) Business Rules
IT professional	Installs a Web browser
IT professional	Tests connections to the Web and network

Setting Up Applications

Subtopics

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

User	Task
Administrator	Designs and creates applications
Administrator	Defines metadata (currencies, scenarios, versions, calendar)
Administrator	Enters exchange rates
Administrator	Defines attributes and attribute values
Administrator	Loads custom dimensions

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 Hyperion Enterprise Performance Management System User and Role Security Guide*. Within Planning, you assign users or groups to secured objects.

Table 4 Setting Up Users and Access Permissions

User	Task
IT professional	Sets up Planning users with a supported user directory
Administrator	Synchronizes users in the user directory with a Planning application
Administrator	Assigns access permissions to users and groups
Administrator	Assigns access permissions to dimensions, members, data forms, folders, and task lists in Planning (see Chapter 3, “Setting Up Access Permissions”)

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 5 Generating Databases

User	Task
Administrator	Creates and refreshes Planning applications based on metadata stored in the application’s relational database

Designing Data Forms

Table 6 Designing Data Forms

User	Task
Administrator and interactive user	Creates and maintains data forms in Planning
Administrator and interactive user	Creates and manages folders in Planning for data form management
Administrator	Assigns access permissions to data forms and folders

Designing Worksheets

Planning users can install Smart View to work with data 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 7 Populating Applications with Data

User	Task
Administrator	Use Copy Data to copy plans from one dimensional intersection to another, including relational data and supporting detail
Administrator and interactive user	Performs bulk loads
Any user	Enters values for global saved assumptions referenced in business rules
Any user	Enters data into the Planning application through data forms or Smart View
Any user	Calculates and creates data using business rules or Copy Version

Creating Business Rules

Table 8 Creating Business Rules

User	Task
Administrator	Assigns users access permissions to business rules
Administrator and interactive user	Creates business rules. See the *Oracle Hyperion Enterprise Performance Management System User and Role Security Guide* for information on Calculation Manager roles and rights.
Administrator, interactive user, and planners who are assigned launch access permissions by an administrator	Launches business rules for Planning applications

For information on administering business rules, see:

- For business rules created with Business Rules: *Hyperion Business Rules Administrator's Guide*.
- For business rules created with Calculation Manager: *Oracle Hyperion Enterprise Performance Management Architect Administrator's Guide*.

See also: "Using Business Rules" on page 140.

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

User Type	Task
Administrator	Adds a standard target-type version
Administrator	Assigns users access permissions (typically, read-only) to the target version
Administrator	Configures the Essbase database so target data is not replaced by lower-level #MISSING values
Any user	Creates target data
Administrator	Publishes targets using Financial Reporting

User Type	Task
Any user	Displays targets on data forms for guidance or input

Creating Tasks Lists

Table 10 Task Lists

User Type	Task
Administrator	Creates task lists that structure the budgeting process for users
Administrator and Task List Access Manager	Assigns access permissions to task lists and tasks

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

User Type	Task
Administrator and interactive user	Creates and launches reports using Financial Reporting
Any user	Prints reports
Any user	Views reports throughout the planning cycle

Initializing Planning Cycles

Table 12 Initializing Planning Cycles

User Type	Task
Administrator	Selects planning units for iterative review, analysis, and approval
Any user	Specifies whether to receive e-mail after becoming planning unit owners

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 data forms. Users read data form instructions and review historical or target data for guidance when preparing plans. For additional user tasks, see the *Oracle Hyperion Planning User's Online Help*.

Users can save data in data forms, and can refresh currently stored data values in data 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 data forms are calculated and stored if the Calculate Data Form business rule is selected to run on save when data 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 Calculation Manager business rules from within Planning (see Chapter 3, “Setting Up Access Permissions”). For more information on business rule access permissions, also see:

- For Calculation Manager: *Oracle Hyperion Enterprise Performance Management System User and Role Security Guide*
- For Business Rules: *Hyperion Business Rules Administrator's Guide*
- *Oracle Hyperion Enterprise Performance Management System User and Role Security 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 e-mail 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 e-mail 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 data 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 13 Starting the Budget Review Process

Users	Tasks
Administrators	Starts the planning unit for the planning cycle
Users with access	Enters data
Users with access	Calculates data
Users with access	Optionally, creates a copy of submissions
Users with access	Promotes data for review
Users with access	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.
Users with access	Notifies reviewers that the planning unit is ready for review
Users with access	Reviews data in the promoted planning unit
Users with access	Reads and modifies promoted data through data forms
Users with access	Runs business rules
Users with access	Reads promoted data using Financial Reporting, Oracle's Hyperion® Web Analysis, Smart View, or third-party reporting tools
Users with access	Modifies data values, enters annotations, signs off on planning units, and promotes, rejects, and approves planning units
Administrators	Checks the planning process status
Administrators	Reads supporting annotations
Administrators	Views the audit trail
Administrators	Completes the planning cycle
Administrators	Changes user access of scenarios and versions in Essbase to read-only
Administrators and interactive users	Publishes reports using Financial Reporting

Post-Planning Activities

Table 14 Post-Planning Activities

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

2

Getting Started with Planning

In This Chapter

Overview

Planning applications are 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 Classic 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 Classic application administration tasks, including:

- Create, delete, and register Classic applications.
- Create data forms, task lists, and menus.
- Assign access permissions for dimension members, task lists, data forms, and data form folders. See Chapter 3, "Setting Up Access Permissions" and the *Oracle Hyperion Enterprise Performance Management System User and Role Security Guide* and its information on Planning.
- 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 Hyperion Enterprise Performance Management Architect 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 75.

If you are not using Performance Management Architect, you work with applications using Classic application administration. If you upgrade an application created in Classic application administration to Performance Management Architect, you cannot return to working with that application in Classic 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 Classic applications). To create and update UDAs, see "Working with UDAs" on page 312.

Because member formula support is available in Performance Management Architect and Planning (for Classic applications) 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 Classic applications).

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 Online Help*.

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 83.

Understanding Implied Sharing in Planning Data 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. Essbase 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 data 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 data form is saved. Likewise, if a value is added for the child, the parent usually assumes the same value after a data 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 data forms. See "Implied Sharing Issues" on page 340.

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 Classic 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 Online Help*.

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:**

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

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

Setting Planning 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

PROPERTY_NAME Column	Description
RDB_DRIVER RDB_SERVER_URL	See "Changing the JDBC Driver" on page 36.
JDBC_MIN_CONNECTIONS JDCB_MAX_CONNECTIONS	See "Configuring JDBC Connection Pooling" on page 37.
OFFLINE_COMPRESSION_THRESHOLD	See "Setting Offline Compression" on page 38.
SUPPORTING_DETAIL_CACHE_SIZE	See "Allocating Memory for Supporting Detail Cache" on page 82.
SUBST_VAR_CACHE_LIFETIME	See "About Selecting Substitution Variables as Members" on page 165.
DIRECT_DATA_LOAD DATA_LOAD_FILE_PATH	See "Loading Data" on page 86.
OLAP_PATH_SEPARATOR	See "Configuring Data Synchronization on UNIX" on page 37.
ENABLE_FOR_OFFLINE	See "Setting Data Form Precision, Display Properties, and Other Options" on page 129.
SMART_VIEW_DISPLAY_WARNING	See "Controlling Smart View Messages" on page 38.
RULE_MAX_WAIT, HBR_MAX_WAIT_FOR_RULE, HBR_MONITOR_DELAY, RULE_MONITOR_DELAY, CLR_CELL_MAX_WAIT, CLR_CELL_MONITOR_DELAY, COPY_DATA_MAX_WAIT, COPY_DATA_MONITOR_DELAY, PUSH_DATA_MONITOR_DELAY, PUSH_DATA_MAX_WAIT JOB_STATUS_MAX_AGE	See "Setting Background Processing" on page 39.
CAPTURE_RTP_ON_JOB_CONSOLE	See "About Runtime Prompts" on page 142.
MAX_VALIDATION_RECORDS VALIDATION_CACHE_SIZE	See "Setting Data Validation Properties" on page 40.
MAX_CELL_TEXT, MAX_CELL_NOTE	See "Setting the Maximum Length for Text Values and Comments in Cells" on page 41.

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, 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 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 you create when selecting the Configure Database task under Planning in Oracle's 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

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

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 JDBC connection pool as one, and the maximum as five. Doing this 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 2 and a maximum of 10. The Planning application does not function correctly if you lower the maximum `JDBC_MAX_CONNECTIONS` to less than 2.

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, in Planning, select **Administration**, then **Manage Properties**. Click **Application Properties** and add the value `OLAP_PATH_SEPARATOR=/`. (The default value is `\`, 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**, 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 displays 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.

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 Online Help*).

➤ To set background processing and the frequency of status checks:

1 Select **Administration**, then **Application**, 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**:

Table 17 Job Property Names and Values

PROPERTY_NAME Column	Property Value	Description
HBR_MAX_WAIT_ FOR_RULE	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.	For business rules created with Business Rules or those migrated from Business Rules to this release. 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 data form is loaded or saved never run in the background.
HBR_MONITOR_ DELAY	The default and minimum value is 180000 milliseconds (3 minutes). If you set a lower value, that value is ignored, and the property is set to the default value.	The interval in milliseconds for checking the status of business rules created with Business Rules.
RULE_MAX_WAIT	The default and minimum value is 180000 milliseconds (3 minutes). If you set a lower value, that value is ignored, and the property is set to the default value.	For business rules created with Calculation Manager. 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 data form is loaded or saved never run in the background.
RULE_MONITOR_ DELAY	The default and minimum value is 180000 milliseconds (3 minutes). If you set a lower value, that value is ignored, and the property is set to the default value.	The interval in milliseconds for checking the status of business rules created with Calculation Manager.
CLR_CELL_MAX_ WAIT	The default and minimum value is 180000 milliseconds (3 minutes). If you set a lower value, that value is ignored, and the property is set to the default value.	The interval in milliseconds to wait before running Clear Cell Details jobs in the background. If this property is not set, Clear Cell Details jobs do not run in the background.

PROPERTY_NAME Column	Property Value	Description
CLR_CELL_ MONITOR_DELAY	The default and minimum value is 180000 milliseconds (3 minutes). If you set a lower value, that value is ignored, and the property is set to the default value.	The interval in milliseconds for checking the status of Clear Cell Detail operations.
COPY_DATA_MAX_ WAIT	The default and minimum value is 180000 milliseconds (3 minutes). If you set a lower value, that value is ignored, and the property is set to the default value.	The interval in milliseconds to wait before running Copy Data jobs in the background. If this property is not set, Copy Data jobs do not run in the background.
COPY_DATA_ MONITOR_DELAY	The default and minimum value is 180000 milliseconds (3 minutes). If you set a lower value, that value is ignored, and the property is set to the default value.	The interval in milliseconds for checking the status of Copy Data jobs.
PUSH_DATA_ MONITOR_DELAY	The default and minimum value is 180000 milliseconds (3 minutes). If you set a lower value, that value is ignored, and the property is set to the default value	The interval in milliseconds for checking the status of Push Data jobs.
PUSH_DATA_MAX_ WAIT	The default and minimum value is 180000 milliseconds (3 minutes). If you set a lower value, that value is ignored, and the property is set to the default value	The interval in milliseconds to wait before running Push Data jobs in the background. If this property is not set, Push Data jobs do not run in the background.
JOB_STATUS_ MAX_AGE	Specify the value in milliseconds. There is no default or minimum value.	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.

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

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 data 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**, 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 data form cells as described in the *Oracle Hyperion Planning User's Online Help*. By default, the maximum number of single-byte characters allowed for text in each cell is 255, and the maximum number for comments is 1500. Note that, in the database, the column data type is set to varchar(255) and varchar(2000) 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 2000 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.**

Logging On to EPM Workspace and Accessing Planning

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

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

➤ To log on to EPM Workspace and access Planning:

1 **Ensure that the Web server is started and the Web application server is running in the Services panel.**

2 **In your browser, enter the URL for the EPM Workspace Log On page.**

3 **Enter your user name.**

4 **Enter your password.**

5 **Click Log On.**

6 **In EPM Workspace, select Navigate, then Applications, and then Planning. Select a Planning application. If prompted, enter your logon information. Planning does not support non-ASCII characters for passwords.**

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

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

You can use the following utilities with Planning.

Table 18 Planning Utilities

Utility	Description	See
BroadcastMessage	Use broadcast messaging to communicate a text message to all Planning users currently logged on to an application.	"Using Broadcast Messaging" on page 74
CalcMgrCmdLineLauncher	Launch business rules or rulesets created with Calculation Manager.	"Launching Business Rules With a Utility" on page 146
CubeRefresh	Create and refresh application databases used to store data in Essbase for each plan type in the application.	"Working with Classic Applications" on page 305
DeleteSharedDescendant	Delete shared dimension members that are descendants of a given member.	"Working with Classic Applications" on page 305
ExportSecurity	Export Planning access permissions to a file, enabling you to export and import access permissions across applications.	"Exporting Access Permissions" on page 60
FormDefUtil	Move data form definitions between Planning applications by exporting or importing to or from an XML file.	"Importing and Exporting Data Forms" on page 155
HBRMigrateSecurity	Migrate launch access permissions on business rules and their projects from Business Rules to Calculation Manager business rules in Planning.	"Migrating Business Rule Security" on page 70
HspUnlockApp	Clear all records in the HSP_LOCK table.	"Unlocking Applications" on page 73

Utility	Description	See
ImportFormDefinition	Import a data form definition from a text file into a Planning data form.	“Overview of Importing Data Form Definitions” on page 152
ImportSecurity	Load access permissions for users or groups from a text file into Planning.	“Importing Access Permissions” on page 57
Outline Load	Load metadata and data for Planning applications.	“Working with the Outline Load Utility” on page 86
PasswordEncryption	Enable suppressing password prompts when running Planning utilities that prompt for passwords.	“Suppressing Password Prompts in Planning Utilities” on page 44
ProvisionUsers	Synchronize Planning users, groups, and roles in the Shared Services Console with a Planning application and with Essbase.	“Synchronizing Users With the Provision Users Utility” on page 68
PushData	Schedule pushing data to a reporting application.	“Pushing Data with a Utility” on page 238
SortMember	Sort dimension members for Entity, Account, Scenario, Versions, and user-defined custom dimensions.	“Working with Classic Applications” on page 305
TaskListDefUtil	Move task list definitions between Planning applications by exporting or importing task list definitions to or from an XML file.	“Importing and Exporting Task Lists” on page 222
UpdateNativeDir	Update identities in Planning with the changes in Oracle's Hyperion® Shared Services.	“Synchronizing Users and Groups With the UpdateUsers Utility” on page 67
UpdateUsers	Synchronize user and group identities between Planning and Shared Services.	“Synchronizing Users and Groups With the UpdateUsers Utility” on page 67

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 45).

➤ 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 Hyperion 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.

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 Hyperion Enterprise Performance Management System Installation and Configuration Guide* or Online Help.

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**.

3

Setting Up Access Permissions

In This Chapter

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 Hyperion Enterprise Performance Management System User and Role Security Guide* and its information on Planning.
- 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 Classic applications. For Performance Management Architect, allows access to application

administration options for Planning. For Classic, allows access to the Classic Application Administration options for Planning (in combination with the Planning Application Creator role).

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

- User-defined dimensions. Assign access permissions to members by selecting the dimension property **Apply Security**. If you omit setting **Apply Security**, all users can access the dimension's members. By default, the Account, Entity, Scenario, and Version dimensions are enabled for access permissions.
- Users and groups, which can vary among applications. Assign access to Planning application elements by using **Assign Access**.

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

Planning Elements 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 Calculation Manager business rules
- Data forms
- Data form folders and Calculation Manager 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 Smart View and Essbase 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, data 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 that 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 may determine 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

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

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. For interactive users and planner user types, access permissions that are specifically assigned to users.

3. Access assignments that are acquired by belonging to a group.
4. 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 **Click Edit.**

4 **In Dimension Properties, select Apply Security to allow access permissions to be set on its members.**

If you do not select this option, there is no security on the dimension, and users can access its members without restriction.

5 **Click Save.**

Click **Refresh** 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 50.

➤ To assign access to 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 **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 **Add, change, or remove access.**

See "Adding Access" on page 51, "Editing Access" on page 51, and "Deleting Access" on page 52.

Adding Access

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

➤ To assign access permissions 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 Calculation Manager business rules:** Select **Administration**, then **Business Rule Security**. Select the folder containing the business rules, then select the business rules.

2 **Click Assign Access.**

- **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**.
- **Optional:** To remove deprovisioned or deleted users or groups from the Planning database to conserve space, click **Remove Nonprovisioned Users/Groups**.

3 **Click Add Access.**

4 **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.

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

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

6 **For Calculation Manager 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 **For the selected users or groups, select the access type and click Add.**

8 **Click Close.**

Editing Access

You can access the **Edit Access** page for a member by clicking **View** in the **Security** column on the **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 Click **Assign Access**.

- **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**.
- **Optional:** To remove deprovisioned or deleted users or groups from the Planning database to conserve space, click **Remove Nonprovisioned Users/Groups**.

3 Click **Edit Access**.

4 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.

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

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

6 Click **Set**.

7 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 Click **Assign Access**.

- **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**.
- **Optional:** To remove deprovisioned or deleted users or groups from the Planning database to conserve space, click **Remove Nonprovisioned Users/Groups**.

3 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.

4 **Click Remove Access.**

5 **Click OK.**

6 **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 75).

Sample Entity Members

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

Table 20 Example of Inherited Access to Shared Members

Case	Access Permission	Effective Access for Base and Shared Member CA	Explanation
Case 1	CA (base) = None iDescendants (West) = Read	Read	CA inherits Read access from its West parent because Read is less restrictive than None.

Case	Access Permission	Effective Access for Base and Shared Member CA	Explanation
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 Data Forms and Folders

Assigning Access to Data Forms and Folders

Administrators can assign access to data forms, data 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 48 and "Assigning Access to Members and Business Rules" on page 50.)

Principles:

- **Data forms:**
  - Planners and interactive users can view or enter data only into data forms to which they have access (and can work only with members to which they have access).
  - Administrators and interactive users can design data forms.
  - Interactive users can access data forms they created or to which an administrator assigned them access.
  - Administrators have write access to all dimension members and to all data forms.
- **Calculation Manager business rules**: Planners can see and launch only business rules to which they are assigned Launch access.
- **Data form folders and Calculation Manager folders:**
  - Planners who are assigned access to a data form folder can access the data 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 data 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 data form folder called Folder1 that contains a data 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 Data Forms and Folders" on page 55.

Adding Access to Data Forms and Folders

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

1 **Select the data form or folder.**

- For data forms and folders, see "Selecting and Opening Data Forms and Folders" on page 124.
- 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 data form, business rule, or folder at a time.

2 **Click Assign Access.**

- **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**.
- **Optional:** To remove deprovisioned or deleted users or groups from the Planning database to conserve space, click **Remove Nonprovisioned Users/Groups**.

3 **Click Add Access, and select the users or groups to access the data form or folder.**

- Click **Users** to display all user names; click **Groups** to display all groups.
- If there are multiple pages of users and groups, type the page number to go to in **Page**, and click **Go**.
- Click **Start** or **End** to navigate to the first or last page.
- Click **Prev** or **Next** to move to the previous or next page.

4 **For Type of Access, select the kind of access users or groups have to the data form or folder.**

For business rules or their folders only:

- Click **Launch** to allow the selected users and groups to launch the selected business rules.
- Click **No Launch** to prevent the selected users and groups from launching the selected business rules.

5 **Click Add.**

6 **Click Close.**

Changing Access to Data Forms and Folders

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

1 **Select the data form or folder.**

- For data forms and folders, see "Selecting and Opening Data Forms and Folders" on page 124.
- For Calculation Manager business rule folders, select **Administration**, then **Manage**, then **Business Rule Security**.

2 **Click Assign Access.**

- **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**.
- **Optional:** To remove deprovisioned or deleted users or groups from the Planning database to conserve space, click **Remove Nonprovisioned Users/Groups**.

3 **Select the users or groups for which to change access, and click Edit Access.**

Click **Users** to display all user names; click **Groups** to display all groups.

4 **For Type of Access, select the kind of access users or groups have to the data form or folder.**

5 **Click Set.**

6 **Click Close.**

➤ To remove access from data forms or folders:

1 **Select the data form or folder.**

- For data forms and folders, see "Selecting and Opening Data Forms and Folders" on page 124.
- 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.**

- **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**.

- **Optional:** To remove deprovisioned or deleted users or groups from the Planning database to conserve space, click **Remove Nonprovisioned Users/Groups**.

3 **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.

4 **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 Hyperion Enterprise Performance Management System User and Role Security Guide*.) Importing access permissions overwrites existing access assignments only for imported members, data forms, data 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 60.

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 45).
- 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 50).
- 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.

Item	Description

username or group name	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 `SecFile.txt` file that pertains to the group: `sl_group`
	For example:
	`admin,member1,READ,MEMBER`
	`admin,member1,READ,MEMBER,SL_GROUP`

artifact name	The named artifact for the imported access permissions (for example the member, data 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"`.

access permissions	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.

Essbase *access flags*	@CHILDREN, @ICHILDREN, @DESCENDANTS, @IDESCENDANTS and MEMBER. Security implementation for these functions is identical to Essbase. **Note:** For task lists, only MEMBER can be used. For folders, only @IDESCENDANTS can be used.
artifact type	For artifacts other than members, distinguish which artifact you are importing security for with *artifact type* identifier: ● `SL_FORM`—for data forms ● `SL_COMPOSITE`—for composite data forms ● `SL_TASKLIST`—for task lists ● `SL_CALCRULE`—for Calculation Manager business rules (not business rules in Business Rules) ● `SL_FORMFOLDER`—for data 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

User3,TaskList09,READWRITE,MEMBER,SL_TASKLIST

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

➤ To import access permissions into Planning:

1 Locate the `ImportSecurity` utility by navigating to the `planning1` directory (for the full path, see "About EPM Oracle Instance" on page 45).

2 From the Command Prompt, enter this case-sensitive command, one space, and the parameters, separating each with a comma. Enclose the parameters with double quotation marks:

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

where:

Parameter	Description
`[-f:`*`passwordFile`*`]`	**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 44.
`appname`	Name of the Planning application to which you are importing access permissions.
`username`	Planning administrator user name.
`delimiter`	**Optional**: SL_TAB, SL_COMMA, SL_PIPE, SL_SPACE, SL_COLON, SL_SEMI-COLON. If no delimiter is specified, comma is the default.
`RUN_SILENT`	**Optional**: Execute the utility silently (the default) or with progress messages. Specify 0 for messages, or 1 for no messages.
`[SL_CLEARALL]`	**Optional**: Clear existing access permissions when importing new access permissions. Must be in uppercase.

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 45.

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 57). 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, data forms, data form folders, task lists, Calculation Manager business rules, and Calculation Manager business rule folders. `ExportSecurity` appends an artifact type flag that specifies whether the exported artifact security is for a data form, composite data form, data form folder, task list, Calculation Manager business rule, or Calculation Manager 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 45).**

2 **From the Command Prompt, enter this case-sensitive command, one space, and the parameters. Separate each parameter with a comma:**

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

where:

Parameter	Description	Mandatory?
`[-f:`*`passwordFile`*`]`	**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 44.	No

Parameter	Description	Mandatory?
`/A=`*`appname`*	The name of the Planning application from which you are exporting access permissions.	Yes
`/U=`*`username`*	The administrator's ID for logging into the application.	Yes
`/S=`*`searchCriteria`*	The user or group name. You cannot use this option with `/S_USER` or `/S_GROUP`.	No
`/S_USER=`*`user`*	A specified user name. You cannot specify multiple users or use this option with `/S_GROUP` or `/S=`*`searchCriteria`*.	No
`/S_GROUP=`*`group`*	A specified group. Only matching groups, not matching user names, are exported. You cannot specify multiple groups or use this option with `/S_USER` or `/S=`*`search criteria`*.	No
`/S_MEMBER=`*`MemberName`*	A specified member. You can specify only one member-based parameter.	No
`/S_MEMBER_ID=`*`MemberName`*	A specified member and its descendants.	No
`/S_MEMBER_D=`*`MemberName`*	A specified member's descendants.	No
`/S_MEMBER_IC=`*`MemberName`*	A specified member and its children.	No
`/S_MEMBER_C=`*`MemberName`*	A specified member's children.	No
`/DELIM=`*`delim`*	SL_TAB, SL_COMMA, SL_PIPE, SL_SPACE, SL_COLON, SL_SEMI-COLON. If no delimiter is specified, comma is the default.	No
`/DEBUG=`	Specify `true` to display the utility's performed steps. `false` is the default.	No
`/TO_FILE=`	Specify the path to the `SecFile.txt` file. By default, the file is in the `planning1` directory (for the full path, see "About EPM Oracle Instance" on page 45). If you specify another path, use double backslashes, for example: `C:\\Oracle\\SecFile.txt`.	No
`/HELP=Y`	Specify as the only parameter to display the syntax and options for `ExportSecurity`.	No

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

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

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

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

3 **If prompted, enter your password.**

Also note:

- If a member, user, or group name contains a character used as the delimiter, the name is enclosed in double 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:

Item	Description
`user` or *`group`*	The name of a user or group defined in Shared Services Console.
`memName`	A member in the application.
`access permissions`	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.
Essbase *`access flags`*	`@CHILDREN`, `@ICHILDREN`, `@DESCENDANTS`, `@IDESCENDANTS`, and `MEMBER`. Security implementation for these functions is identical to Essbase.
`artifact type`	After each line, the utility appends the artifact type: ● `SL_FORM`—for data forms ● `SL_COMPOSITE`—for composite data forms ● `SL_TASKLIST`—for task lists ● `SL_CALCRULE`—for Calculation Manager business rules (not business rules in Business Rules) ● `SL_FORMFOLDER`—for data form folders ● `SL_CALCFOLDER`—for folders containing Calculation Manager business rules **Note:** If you manually create the `SecFile.txt` file, you must add the `artifact type` identifiers.

For example, an exported file might contain these lines:

```
User1,DataForm2,READ,MEMBER,SL_COMPOSITE

User2,Folder3,READWRITE,MEMBER,SL_FORMFOLDER

User3,DataForm4,READWRITE,MEMBER,SL_FORM

”North America”,Account101,READWRITE,MEMBER,SL_CALCFOLDER
```

Reporting on Access Permissions

Subtopics

- Selecting Reporting Objects
- Selecting Reporting Options
- Working With Access Permissions Reports
- Setting up Audit Trails

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 data forms.

➤ To select reporting objects:

1 Start the Access Control Report.

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.**

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 21 Access Report Types

Report Type	Description	Options
Assigned Access	Summarizes access permissions that administrators assign	Specify whether access is assigned by member selection relation or group membership: ● **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.

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 data forms, business rules, approvals, users, access permissions, and so on. Access assignments imported with the `ImportSecurity` utility are not reflected in audit reports. To view audits, administrators create and run reports using RDBMS report writers.

Table 22 Actions That Can be Audited

Audit Options	Tracked Changes
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 data 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)
Data Form Definition	Data forms: creating, modifying, adding rows. (The audit record does not record how the design changed.)
Data 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, data forms, data 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	Data forms taken offline or synchronized back to the server
Task List	Task lists: created, copied, saved, moved, and deleted
Copy Data	Users' selections for Static Dimensions, Source Dimension, and Destination Dimension
Clear Cell Details	Users' selections for clearing supporting detail, cell text, and account annotations

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

1 Select **Administration**, then **Application**, 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 Web 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:

Tracked Changes	Examples
The type of change	Metadata, data, data form, access permissions, planning units
The affected object (The columns ID_1 and ID_2 in the audit report help define the object that changed.)	Data Form: Expenses 04 Group: Marketing
User	VHennings
Time Posted	12/22/2011 8:17
Action	Add
Property	Currency
Old value	Default
New value	USD

Viewing and Clearing Audit Reports

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

```
DELETE FROM HSP_AUDIT_RECORDS
```

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

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

Managing Security Filters

Access permissions in 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 in Essbase 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 75). First validate that the size of security filters does not exceed the Essbase limit of 64 KB per row.

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.

Note: 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.

Synchronizing Users and Groups With the UpdateUsers Utility

If you move users and groups, their identities (SIDs) change, and unless you synchronize their identities between Planning and Shared Services, their assigned access permissions are lost. Situations requiring synchronization:

- You change authentication providers.
- Users, groups, or organizational units (OUs) are moved in an external provider.
- You migrate your application from one environment to another (for example, from Dev to Prod) and plan to change Shared Services repositories.

You can use two utilities in concert to synchronize changes:

1. Run the `UpdateNativeDir` utility to update user and group identities in Shared Services.

   `UpdateNativeDir` and its documentation are in the *`EPM_ORACLE_INSTANCE`*`/common/utilities/UpdateNativeDir` directory.

   For more information on the directory path, see "About EPM Oracle Instance" on page 45.

2. Run the `UpdateUsers.cmd` utility to update the SIDs in Planning with the changes in Shared Services.

Note: It is important to run the utilities in this order.

➤ To launch the `UpdateUsers` utility:

1 **At the Command Prompt, from the `planning1` directory, enter the command, using this syntax:**

```
updateusers [-f:passwordFile] serverName  adminName applicationName
```

For the full path to `planning1`, see "About EPM Oracle Instance" on page 45.

Table 23 UpdateUsers Syntax

Parameter	Description	Required?
`[-f:passwordFile]`	**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 44.	No
`serverName`	The server on which the Planning application resides	Yes
`adminName`	The administrator's name for logging on to the Planning application.	Yes
`applicationName`	The Planning application for which to synchronize users and groups (must be on the server on which the utility is run).	Yes
`/?`	Specified by itself, prints the syntax and options for `UpdateUsers`.	No

For example:

```
updateusers.cmd ABCserver admin Planningapp
```

2 **If prompted, enter your password.**

3 **Optional: To view information on the results, open the `UpdateUsers.log` file in the *`EPM_ORACLE_INSTANCE`*`/diagnostics/logs/planning` directory.**

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 45.

Table 24 ProvisionUsers Syntax

Parameter	Description	Required?
`[-f:`*`passwordFile`*`]`	**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 44.	No
`/ADMIN:`*`adminName`*	The administrator's name for logging on to the Planning application.	Yes
`/A:`*`appName`*	The Planning application to synchronize (must be on the server on which the utility is run).	Yes
`[/U:`*`user1[;user2;user3]`*`]`	Specifies users to synchronize. For example, to synchronize users Planner1 and Planner2, use `/U:Planner1;Planner2`. Omitting this argument synchronizes all users.	No
`[/R:`*`n`*`]`	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.	No
`/?`	Specified by itself, prints the syntax and options for `ProvisionUsers`.	No

2 **If prompted, enter your password.**

Example 1

Entering:

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

Synchronizes all users in the App1 application.

Example 2

Entering:

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

Synchronizes user Planner1 in the App2 application every 60 minutes.

Migrating User and Group Identities

When you change a user or group's identity or their position in the user directory hierarchy, you must update—or migrate—this information to 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 **Data Forms and Ad Hoc Grids**, and then select a data 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 application formerly used Business Rules, but now Calculation Manager is the selected calculation module for your application, 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`**:**

- If you are using Classic application administration, ensure that Calculation Manager is the selected calculation module for the application (see "Specifying System Settings" on page 231). If you upgraded the application to Performance Management Architect, then Calculation Manager is automatically selected as the calculation module.
- Migrate business rules from Business Rules to Calculation Manager. See the *Calculation Manager Designer's Guide*.
- Deploy the business rules to Planning.

2 **At the command line, from the** `planning1` **directory, enter this command and its parameters, separating each by a space:**

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

For the full path to `planning1`, see "About EPM Oracle Instance" on page 45.

Table 25 HBRMigrateSecurity Parameters

Parameter	Purpose	Required?
`[-f:passwordFile]`	**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 44.	No
`/A:appname`	Specify the Planning application to which to migrate launch access permissions for business rules	Yes

Parameter	Purpose	Required?
`/U:`*`admin`*	Specify the administrator's user name	Yes
`/F:`*`output file`*	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 45). This file contains a log of the transaction, and helps with troubleshooting.	Yes
`/?`	Print the syntax and options for `HBRMigrateSecurity.cmd`	No

3 **If prompted, enter your password.**

For example:

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

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 deprovisioned users and groups 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 **Data Forms and Ad Hoc Grids**, and then select a data form folder or data 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

4 Managing Planning Databases

In This Chapter

For Classic Planning applications, 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*.

Opening Applications

You can open applications to work with data or run reports. Multiple applications can reside on the same server and be open at the same time. This can slow performance, depending on the server's configuration, the processor speed of your computer, and available memory. To determine server configuration, contact your Oracle consultant.

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 45.

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 44.

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

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, whether you send the broadcast message through the Web or the command line. 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 using Planning Web:

1 **Select Administration, then Application, then Broadcast Message.**

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 45.

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 44.

`SERVER_NAME`: the localhost name.

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

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

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

3 **If prompted, enter your password.**

4 **You can view status, errors, or information for broadcasting messaging in the console.**

For example:

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

Viewing Usage Statistics

You can determine which 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 82).

➤ 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 Essbase for each plan type in the application. Databases are structured according to dimensions, hierarchical members, attributes, and other data specified in applications. Essbase creates an encrypted data file (`essbase.sec`) to store access permission information.

When you create an application, select **Create** to update the Essbase 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 applications are not reflected to users performing data entry and approvals tasks until you refresh the Essbase 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:

- Essbase security filters are updated.
- Currency conversion calc scripts are updated.
- Members and associated properties are propagated from the relational database to the Essbase database.
- Custom attributes are added, modified, or deleted in the Essbase database.
- Exchange rate values are repopulated in the Essbase 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 Essbase database is restructured.
- UDAs are added to the Essbase database.

Caution! Oracle recommends backing up the application before creating or refreshing. See "Backing Up Applications and Application Databases" on page 83. 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 32.

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 Essbase 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 Essbase application is refreshed. Planning does not log out users during refresh. See "Limiting Use of Applications" on page 232.

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 83.**

2 Select **Administration**, and then **Application.**

3 Select **Create Database** or **Refresh Database.**

4 Select options:

- **Database**: Creates or refreshes an Essbase 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 Online Help*.
- **Security Filters**: Generates security filters in Essbase for use by third-party applications. Stores access permissions in an encrypted data file (`essbase.sec`). 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 66.

  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 Applications" on page 232.

- **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 53).

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

  When this option is cleared, each base and shared member instance uses the member's own security. This means the same cell value could have Read access in one data form and Write access in another data form if each member were selected differently in the respective data forms.

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

Managing Exchange Rates

Exchange rate tables enable budget preparers to create plans in different currencies. For example, you can specify Yen as the base currency for the Japan entity and US dollars for the United States entity. When you display a data 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. (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*.) You can set up exchange rates using the procedure in "Creating Exchange Rate Tables" on page 78 and "Editing Exchange Rate Tables" on page 79.

Multiple-currency applications store 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 Dense. You can change this 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**
4. Click **Save**, then define settings for the table as specified in "Editing Exchange Rate Tables" on page 79.

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 26 Exchange Rate Table Options

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

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

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

5. Click **Save.**

Deleting Exchange Rate Tables

➤ To delete exchange rate tables:

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

Managing Currency Conversion

➤ To manage currency conversion:

1. Select **Administration**, then **Manage**, and then **Currency Conversion.**
2. Click **Create.**
3. In the **Create File** tab, specify information for the Currency Conversion Script file, then click **Next.**
   - **Name**
   - **Description**
4. 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 Essbase 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 Essbase 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 264, “About Reordering Dimensions” on page 82, and “Setting Dimension Density and Order” on page 276).

- Design business rules to execute within two minutes. Schedule longer-running business rules to execute in the background (see "Setting Background Processing" on page 39).
- Set the **Suppress Missing Blocks** option (see "Setting Data Form Grid Properties" on page 127).
- 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 Planning Properties" on page 35).
- Optimize the JDBC connection pool settings (see "Configuring JDBC Connection Pooling" on page 37).
- Increase the timeout value for SQL queries (see "Optimizing SQL Relational Databases" on page 34).
- If you use the WebLogic Server, check its performance settings (see "Optimizing WebLogic Server Parameters" on page 343).
- Tune Java Virtual Machine (JVM) parameters (see "500 Error Message" on page 340).
- Optimize Windows network parameters (see "Optimizing Windows Network Parameters" on page 344).

About Reordering Dimensions

The order of dimensions is critical for the structure and performance of Essbase databases. 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 75. To change its value, see "Setting Planning Properties" on page 35.

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 Online Help*.

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 data forms into multiple smaller data forms having fewer rows and columns. Organize related data forms into folders.
- Using account annotations impacts performance, so use them sparingly.
- The first time Planning is launched and data forms are opened, caches are loaded, which takes more time than in subsequent sessions. So, before your users start using Planning, launch the product and open the most commonly used data 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 Hyperion Enterprise Performance Management System Backup and Recovery Guide*.

5 Loading Data and Metadata

In This Chapter

Overview of Loading Information

You can use these tools to load metadata and data as described in the following tables.

Table 27 Metadata Load Tools

Metadata Load Tool	For Classic Planning	For Performance Management Architect	Comments
Outline Load utility	X		See "Working with the Outline Load Utility" on page 86.
Oracle Data Integrator (ODI) Adapter for Planning	X		See the *Hyperion Data Integration Management Adapter for Planning User's Guide*.
Data Integration Management (DIM) Adapter for Planning	X		See the *Oracle Data Integrator Adapter for Planning Online Help*.
Oracle Hyperion Financial Data Quality Management ERP Integration Adapter for Oracle Applications	X	X	Loads from supported ERP Integrator source systems. See the *Oracle Financial Data Quality Management ERP Integration Adapter for Oracle Applications Administrator's Guide*.
Performance Management Architect flat files		X	See the *Oracle Hyperion Enterprise Performance Management Architect Administrator's Guide*.
Performance Management Architect interface tables		X	See the *Oracle Hyperion Enterprise Performance Management Architect Administrator's Guide*.

Table 28 Data Load Tools

Data Load Tool	For Classic Planning	For Performance Management Architect	Comments
Outline Load utility	X		Loads numeric, date, and text data values. See "Working with the Outline Load Utility" on page 86.
Performance Management Architect data synchronization		X	Loads data for Planning. See the *Oracle Hyperion Enterprise Performance Management Architect Administrator's Guide*.
Essbase flat file load	X	X	Loads numeric data for Planning. See the *Oracle Hyperion Enterprise Performance Management Architect Administrator's Guide*.
ERP Integrator	X	X	Loads from supported ERP Integrator source systems. See the *Oracle Financial Data Quality Management ERP Integration Adapter for Oracle Applications Administrator's Guide*.
Oracle Hyperion Financial Data Quality Management Adapter for Planning, Fusion Edition (FDM)	X	X	Loads numeric data for Planning. See the *Hyperion Financial Data Quality Management Administrator's Guide*.

Working with the Outline Load Utility

The Outline Load utility can be used to load metadata and data for Account, Period, Year, Scenario, Version, Currency, Entity, user-defined dimensions, attributes, UDAs, exchange rates, Smart Lists, and planning unit hierarchies.

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. Create a load file for each dimension or set of data that you want to load.
3. Test the load files, and then run the utility.

   For detailed instructions, see these topics:

   - "Loading Data" on page 86 and "Loading Metadata" on page 90
   - "Command Line Parameters for the Outline Load Utility" on page 91
   - "Dimension Properties" on page 98

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 data form. For example, if Jan or Descendants (YearTotal) are columns in a data 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.

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 Hyperion 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 Oracle Essbase Administration Services).

     In most cases, set DIRECT_DATA_LOAD to False, and set DATA_LOAD_PATH to the location and name that will be used for the generated data and rules files, for example, C:/myDirectory/App1.txt. Ensure that these properties are set in the **System Properties** tab in the **Manage Properties** page.

   d. Restart the Planning application server.

3 **If you want to load to driver members that are specified in Planning, set the driver members as described in this step. Otherwise, skip to the next step.**

   a. Log on to the Planning application for which data will be loaded.

   b. Select **Administration**, 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 data 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 data form.

4 **Generate a comma-separated load file containing these columns:**

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

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

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

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

Example 2: In this example, Entity was selected as the Data Load dimension on the application's **Data Load Administration** page. Account was selected as the Driver Dimension, and Account members aUnspec, aSmart, aDate, and aText were selected as the driver members. This .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:

```
    Currency  Version  Scenario  Year  Entity  Period  Account  HSP_Rates

77  USD  BUVersion_1  Current  FY07  e1  Jan  aUnspec  HSP_InputValue

1  USD  BUVersion_1  Current  FY07  e1  Jan  aText  HSP_InputValue

20081222  USD  BUVersion_1  Current  FY07  e1  Jan  Date  HSP_InputValue

1  USD  BUVersion_1  Current  FY07  e1  Jan  aSmart  HSP_InputValue
```

Example 3: Specify drivers directly in the .CSV load file.

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

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

sl1_value2,a2,"Jan,Local,e1,Current,Version1,FY08",Plan1

OutlineLoad /A:acpt1 /U:admin /M /I:c:\outline1data.csv /TR /L:c:/OutlineLogs/
outlineLoad.log /X:c:/OutlineLogs/outlineLoad.exc
```

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 45.

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: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
```

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

```
C:\EPM_ORACLE_INSTANCE\Planning\planning1>OutlineLoad /A:test /
U:admin /M /I:c:\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 98.

Loading Metadata

Metadata for Classic Planning applications 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.**

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 98.

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

```
Entity, Parent, Data Storage

e1, Entity,

e2, ,

e1, e2, Shared
```

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

```
Entity

 e1

 e2

  e1(Shared)
```

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

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

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 45.

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 98.

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.

```
OutlineLoad [-f:passwordFile][/S:server] /A:application /U:userName [/
M] [/E:outputFileName] [/I:inputFileName/D[U]:loadDimensionName|/
DA:attributeDimensionName:baseDimensionName] [/TR] [/N] [[/R] [/U]] [/
C] [/F] [/K] [/X:exceptionFileName] [L:logFileName] [/DX:HSP_Rates][/
DS:HSP_SMARTLISTS] [/8][/?]
```

Parameter	Description
`[-f:passwordFile]`	**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 44.
`/S:server`	Server on which the application resides; if not specified, localhost is used.
`/A:application`	Name of the Planning application to which you are importing.

Parameter	Description
/U:*userName*	User name with which to log on to the application.
/M	Generate fully qualified header records for loadable dimensions in the application. Use /-M if you do not want to display this information (default).
/E:*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.)
/I:*inputFileName*	Name of the load file that contains a header record and data records in .CSV format. You must also specify a data load dimension (/D).
/D:*loadDimensionName*	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.
/DU:*userDefinedLoadDimensionName*	User-defined dimension to be loaded; a dimension with this name will be created if it does not exist.
/DA[T]:*attributeLoadDimensionName:baseDimensionName*	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.
/DAN:*attributeLoadDimensionName:baseDimensionName*	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.
/DAB:*attributeLoadDimensionName:baseDimensionName*	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.
/DAD:*attributeLoadDimensionName:baseDimensionName*	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.
/TR	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.
/N	Perform a “dry run” by parsing the load file without loading data or metadata. Use /-N (or do not specify the /N parameter) to parse the load file while loading data and metadata (default).
/TR	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.

Parameter	Description
`/O`	Maintain the order of members in the load file when loading, with the exception of UDAs (default). Use `/-O` to ignore the order of members in the load file when loading.
`/H`	Order input records in parent-child order, with the exception of UDAs (default). Use `/-H` to load input records as they appear in the load file; this option is faster and uses less memory.
`/R`	Delete all members of the load dimension before performing the load. Use `/-R` (or do not specify the `/R` parameter) to keep all members of the load dimension (default). See also `/U`. **Note:** Use caution with `/R`; this option removes attribute bindings and approvals states.
`/U`	Delete all planning units with the `/R` option, or display an error if members in planning units would be deleted. Use `/-U` (or do not specify the `/U` parameter) to prevent deleting members in planning units (default). Use `/U` with `/R` to enable deleting started planning units and deleting all members in the dimension specified in the .CSV load file.
`/T`	Inherit unspecified plan type settings from the parent when adding new members (default). Use `/-T` to force explicit setting of plan type settings for the member.
`/C`	Perform a cube refresh after the metadata load. Use `/-C` if you do not want to perform a cube refresh (default). See also `/F`.
`/F`	Create security filters when refreshing with the `/C` option. Use `/-F` if you do not want to refresh security filters (default). (This option does not provision users to the application; it only creates security filters for users that currently exist. Users can be provisioned to applications using other methods.) For this option to take effect, `/C` must also be specified.
`/K`	Lock the load dimension before loading (default), recommended. Use `/-K` if you do not want to lock the dimension (not recommended unless you are using `/N`).
`/X:`*`exceptionFileName`*	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`.)
`/L:`*`logFileName`*	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`.)
`/DX:HSP_Rates`	Load the HSP_Rates dimension and create exchange rate tables if they do not exist.
`/DS:HSP_SMARTLISTS`	Load the Smart Lists dimension and Smart List dimension entries.

Parameter	Description
/8	Use /8 if the file being exported will contain Unicode characters.
/?	Display usage text.

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

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

```
NumericAttrib,Parent

One,NumericAttrib

1,One

2,NumericAttrib
```

Example: Load Exchange Rates, add EUR as a member of the Currency dimension, and change the year in the .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.

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

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

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

Example: Set Weekly Distribution to Use 445

```
Account, Parent, Use 445

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.

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

Dimension,UDA

Account,New2
```

Example: Load a .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:

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

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.

```
Entity,Parent,Data Storage

A,B,Store

B,e1,Store
```

Example: `/R` parameter

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

```
Entity,Parent,Data Storage,TextAttrib

e1,Entity,Store,

e11,e1,Store,orange
```

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

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

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

Example: `/T` parameter

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

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

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

Example: `/-T` parameter

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

Example: `/TR` parameter

```
OutlineLoad /A:acpt1 /U:admin /M /I:c:\outline1data.csv /TR /L:c:/OutlineLogs/
outlineLoad.log /X:c:/OutlineLogs/outlineLoad.exc
```

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

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

```
sl1_value2,a2,"Jan,Local,e1,Current,Version1,FY08",Plan1
```

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

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

```
SmartList Name, Operation, Label, Display Order, Missing Label, Data Form Setting, 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 data 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.

```
OutlineLoad /S:planqe4 /A:acpt_580 /U:admin /M /E:puh_test2.csv /D:test2

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

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

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

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

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

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

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

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

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

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

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

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

Note: Secondary members for the first four records are not specified.

Dimension Properties

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

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.
- **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 98.

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

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

Column Header in .CSV Load File	Planning Property	Value	Default	Required
Formula	Formula	Text, subject to Essbase restrictions for valid formula syntax; <none>removes a formula	None	No
UDA	UDA	A single UDA text value, or a quoted, comma-separated list of UDA text values; non-existing UDAs are added to the dimension; existing UDA bindings are removed on subsequent saves if not re-specified; nothing specified leaves bindings as is; <none> removes all existing UDA bindings.	None	No
Smart List	Smart Lists	Text, restricted to names of Smart Lists that are already defined for the application. Should be specified only if Data Type is set to Smart List; <none> removes any existing Smart List binding	None	No
Data Type	Data Type	Text: Unspecified, Currency, Non-Currency, Percentage, Smart List, Date, Text	Inherited from the Parent; Currency if the member is added under the root dimension	No
Operation		Text: update; delete level 0; delete idescendants; delete descendants	Update	No
Account Type	Account Type	Text: Expense, Revenue, Asset, Liability, Equity, Saved Assumption	Inherited from the parent; Revenue if the member is added under the root dimension.	No
Time Balance	Time Balance	Text: Flow, First, Balance, Average, avg_actual, avg_365, fill	Inherited from the parent (default Time Balance values for Account types: Expense: FlowRevenue: Flow, , Asset: Balance, Liability: Balance, Equity: Balance)	No
Use 445 Use 544 Use 554 (Only one column header appears, depending on system settings. If weekly distribution is not defined for the application, no column header displays.)	Weekly Distribution (If Weekly Distribution is set to Even for the application, no distribution option displays.)	True, False, or an integer: nonzero is true; zero is false	None	No
Skip Value		Text: None, Missing, Zeros, Missing and Zeros; must be None if Account Type is Expense or Revenue	Inherited from the parent	No

Column Header in .CSV Load File	Planning Property	Value	Default	Required
Exchange Rate Type	Exchange Rate Type	Text: None, Average, Ending, Historical (None should not be specified if Data Type is set to Currency; otherwise, it should be specified)	Inherited from the parent; Average if member is added under root dimension.	No
Variance Reporting	Variance Reporting	Text: Non-Expense, Expense. (Must be Expense if Account Type is Expense; must be Non-Expense for all other Account types.	Inherited from the parent; Non-Expense if the member is added under the root dimension.	No
Source Plan Type	Source Plan Type	Text: Plan Type names defined in the application (for example, Plan1 or Plan2)	Plan1 or the name of the first plan type defined in the application	No
Plan Type (*Plan1*)	Plan Type	True, False, or an integer: non-zero is true; zero is false	Inherited from the parent	No
Aggregation (*Plan1*)	Aggregation	Text: +, –, *, /, %, ~, Never	Inherited from the parent. If the parent is a root member, the default is +; for Year, the default is ~ (ignore)	No
Plan Type (*Plan2*)	Plan Type	True, False, or an integer: non-zero is true; zero is false	Inherited from the parent	No
Aggregation (*Plan2*)	Aggregation	Text: +, –, */%~, Never	Inherited from the parent; if the parent is the root member, the default is +; for Year, the default is ~ (ignore)	No
Plan Type (*Plan3*)	Plan Type	True, False, or an integer: non-zero is true; zero is false	Inherited from the parent	No
Aggregation (*Plan3*)	Aggregation	Text: +, –*/%, ~, Never	Inherited from the parent; if the parent is a root member, the default is +; for Year, the default is ~ (ignore)	No
Attribute Dimension Name (this property is available for sparse Account, Entity, or user-defined dimensions)	Attributes	The name of an attribute defined in the attribute dimension: existing attribute bindings are removed on subsequent saves if not re-specified; nothing specified leaves bindings as is; <none> removes all existing attribute bindings for the member; one column header is displayed for each Attribute dimension defined on the dimension.	None	No

Entity Dimension Properties

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

Notes:

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

Table 30 Entity Dimension Properties

Column Header in .CSV Load File	Planning Property	Value	Default
Entity	Member Name	Text, subject to Planning member naming restrictions.	None
Two Pass Calculation	Two Pass Calculation	`True`, `False`, or an integer: non-zero is true; zero is false (should be set to 1 only if Data Type is `Dynamic Calc` or `Dynamic Calc and Store`)	Inherited from the parent
Base Currency	Base Currency	Text, restricted to currency names already defined in the application	Inherited from the parent; if the member is shared, the default is the base member's currency; if the member is added under the root dimension, the base currency is the default currency defined when the application was created.

Period Dimension Properties

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

Notes:

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

- Plan type names and numbers depend on what is defined in the application.
- For properties common to Period and Account members, see "Account Dimension Properties" on page 100.

Table 31 Period Dimension Properties

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

User-Defined Dimension Properties

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

Plan type names and numbers depend on what is defined in the application. For properties common to User-Defined and Account members, see "Account Dimension Properties" on page 100.

Table 32 User-Defined Dimension Properties

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

Year Dimension Properties

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

Notes:

- The member name for Year must be of the form `FY`*`nn`*.
- 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

Column Header in .CSV Load File	Planning Property	Value	Default	Required
Year	Year member	`FY`*`nn`*	None	Yes

Scenario Dimension Properties

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

Notes:

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

Table 34 Scenario Dimension Properties

Column Header in .CSV Load File	Planning Property	Value	Default	Required
Scenario	Scenario Name	Text	None	Yes
Start Year		First FY year defined in the application (as determined by position)	First year in the application	No
Start Period		First base time period (as determined by position)	First base period in the application	No
End Year		Last FY year defined in the application (as determined by position)	Last year in the application	No
End Period		Last base time period (as determined by position)	Last base period in the application	No

Column Header in .CSV Load File	Planning Property	Value	Default	Required
Exchange Table		The name of an exchange rate table defined in the application.	None	No
Include BegBal		`True`, `False`, or an integer: non-zero is true; zero is false	False	No
Approvals Enabled		`True`, `False`, or an integer: non-zero is true; zero is false	False	No

Version Dimension Properties

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

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

Table 35 Version Dimension Properties

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

Currency Dimension Properties

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

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

Table 36 Currency Dimension Properties

Column Header in .CSV Load File	Planning Property	Value	Default	Required
Currency	Currency Name	Text	None	Yes

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

Attribute Dimension Properties

`Attribute, Parent, Alias: Default, Operation`

Notes:

- For properties common to Attribute and Account members, see "Account Dimension Properties" on page 100.
- **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

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

UDA Dimension Properties

`Dimension, UDA, Operation`

For properties common to UDA and Account members, see "Account Dimension Properties" on page 100.

Table 38 UDA Dimension Properties

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

Exchange Rate Dimension Properties

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

Table 39 Exchange Rate Dimension Properties

Column Header in .CSV Load File	Planning Property	Value	Default	Required
Table	fxTblId	Name of the exchange rates table	None	Yes
Description	Description	Description of the exchange rates table	None	No
To Currency	toCur	Currency defined in the application, to which the conversion will be applied	None	Yes
From Currency	fromCur	Currency defined in the application, from which the conversion will be computed	None	Yes
Operation	N/A	`update` (delete operations are not supported: `delete level 0`, `delete idescendants`, `delete descendants`)	update	No
Method	method	`multiply`; `divide`	multiply	No
Historical	historicalRate	numeric value	0	No
Beg Balance	begBalanceRate	numeric value	0	No

Column Header in .CSV Load File	Planning Property	Value	Default	Required
Year	yearId	A year defined in the application, such as FY08	None	Yes, if Average or Ending is specified
Period	tpId	A base time period defined in the application, such as Jan	None	Yes, if Average or Ending is specified
Average	avgVal	numeric value	None	No
Ending	endVal	numeric value	None	No

Planning Unit Hierarchy Dimension Properties

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

Table 40 Planning Unit Hierarchy Dimension Properties

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

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

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

Smart List Dimension Properties

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

Table 41 Smart List Dimension Properties

Column Header in . CSV Load File	Planning Property	Value	Default	Required
SmartList Name	Name of the Smart List	Text, name of the Smart List/enumeration	None	Yes
Operation	Operation	Text, addsmartlist—creates a new Smart List; addEntry—adds an entry to the Smart List	None	Yes
Label	Label field of Smart List	Text, label field of Smart List	Empty	No
Display Order	Display Order	Integer or text value for display order of the Smart List: `0` or `ID`—order by entry id; `1` or `Name`—order by entry name; `2` or `Label`—order by entry label	ID	No
Missing Label	#Missing Drop Down label	Text, Missing Drop Down label	LABEL_NONE	No
Use Data Form Setting	#Missing Data Form	`True`—Data Form Setting; `False`—Drop Down Setting	Data Form Setting	No
Entry ID	Smartlist entry ID	ID for the Smart List entry	Default entry ID	No
Entry Name	Entry Name	N/A	None	Yes
Entry Label	Entry Label	N/A	None	Yes

Load File Considerations

Consider these points when working with load files:

- 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 98.
- 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 42 Examples of Data Values Containing Commas and Quotation Marks

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

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 91.

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:

```
OutlineLoad /A:pln1dv /U:admin /E:c:/AttribTextColor_export.csv /D:Color /L:c:/
outlineLoad.log /X:c:/outlineLoad.exc
```

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

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

```
Currency, Description, Data Storage

USD,          “Description Line 1

Description Line 2

Description Line 3”, Store
```

```
EUR,        “description for EUR”, Store
```

Considerations for exporting metadata:

- UDAs cannot be exported as a separate dimension. They are exported with the members to which they are assigned, similar to formulas.
- The DTS Period type is recognized, but is not supported by the Outline Load utility. If you try to export DTS information, the utility displays an error in the log file.
- If exported members have aliases in an alias table other than Default and these members are imported into another application, the alias table must be created manually in the application before the members are imported.

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 Data Integrator Adapter for Planning User's Guide*.

Using DIM Adapter for Planning

After installing and configuring Oracle's 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 *Hyperion Data Integration Management Adapter for Planning Online Help*. Oracle's Hyperion® Data Integration Management Adapter for Planning is available only for Classic Planning applications.

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.

6 Click **Save**.

   For detailed instructions on loading, see the Planning adapter documentation described in "Overview of Loading Information" on page 85.

Loading from Administration Services

➤ 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.

```
EUROPE     GROSS SALES     ACTUAL     FINAL     JAN     2008     150

EUROPE     GROSS SALES     ACTUAL     FINAL     FEB     2008     110

EUROPE     GROSS SALES     ACTUAL     FINAL     MAR     2008     200
```

If a dimension uses the same value for all rows, you can place it in the header as a page dimension, as in this example:

```
ACTUAL     FINAL

EUROPE     GROSS SALES     JAN     2008     150

ASIA       NET INCOME      FEB     2008     150

EUROPE     NET INCOME      FEB     2008     110

ASIA       GROSS SALES     JAN     2008     200
```

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* or Online Help.

Loading with FDM

You can load data with Oracle Hyperion Financial Data Quality Management, Fusion Edition. For information, see the *Oracle Hyperion Financial Data Quality Management Administrator's Guide*.

6

Managing Data Forms

In This Chapter

About Data Forms

Subtopics

- Data Forms and Plan Types
- Data Forms and Access Permissions
- Data Forms and Currencies
- Designing Data Forms for Multiple Currencies
- Designing Data Forms for Drill-Through Information
- Designing Data Forms with Formula Rows and Columns
- Designing Data Forms with Data Validation
- Data Forms and Versions
- Data Forms and Attributes
- Data Forms and Shared Members
- Data Forms and Calculations

Data forms are grids for entering data. You can create many data forms to meet your users' needs.

Data Forms and Plan Types

When you create a data form, you associate it with a plan type, which determines the data form's valid members. For example, if you assign a data 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 data form after assigning it.

You can edit data form accounts if their source plan type matches the data form's plan type. If you add an account to a data form for a plan type other than the account’s source plan type, the account is read-only on that data form.

Data Forms and Access Permissions

By assigning access to a data 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 data 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 data forms that include the Europe entity are displayed as read-only. Users can change data only for members to which they have write access.

Data Forms and Currencies

For single-currency applications, all entities use the currency selected when the application was created. For multicurrency applications, the selected Currency member on data 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 data form is read-only. You can enter data in rows or columns that have Currency or Local as the selected member. See "Designing Data Forms for Multiple Currencies" on page 121.

Designing Data 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 data form.
- Assign the Currency dimension to the page axis and select reporting currencies as members to convert currencies for all the members in the data 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 Data Forms for Drill-Through Information

If data forms contain members whose data is loaded from a source such as FDM or ERP Integrator, users can drill through to view more details for the cell data source. To enable data forms for drill through, complete these tasks when designing data forms:

- Within FDM or ERP Integrator, 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 ERP Integration Adapter for Oracle Applications Administrator's Guide*. For the current release, you do not need to set properties or configure Planning for FDM.
- Give appropriate access permissions, for example, to the data 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 ERP Integration Adapter for Oracle Applications. 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 multicurrency applications, 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 data forms that include cells with drill-through information, a drill-through icon is displayed in those cells.

Designing Data 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.

You can include formula rows and columns in data forms, as described in "Adding Formula Rows and Columns" on page 132. For information about creating formulas and using the mathematical functions, see Appendix C, "Data Form Formula Functions."

Designing Data Forms with Data Validation

You can design data 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 data form if entered data violates a validation rule. Data validation rules are saved as part of the data form. See "Including Data Validation Rules in Data Forms" on page 133 and Chapter 8, "Managing Data Validation."

Data 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.

Data 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.

Data 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 data forms.

Data 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.

Understanding Data Forms

Subtopics

- Point of View
- Page Axis
- Rows and Columns

Before creating data forms, understand their components:

- "Point of View" on page 123
- "Page Axis" on page 123
- "Rows and Columns" on page 123

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 data form, in the point of view you can specify only relevant members or include user variables. See "Defining Simple Data Form Page and Point of View" on page 134 and "Managing User Variables" on page 149.

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 Data Form Page and Point of View" on page 134.

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 data forms, they can enter data into the cell where the Unit Sales row intersects with the January column.

By default, data 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 131.

Selecting and Opening Data Forms and Folders

Use this procedure to select and open data form folders and the data forms they contain. After you select a data form folder use the buttons above **Data Form Folders** to create, rename, and assign access to the folder. After displaying a data form, use the buttons above **Data Form** to create, edit, move, delete, and assign access to data forms.

➤ To select and open data forms or data form folders:

1. Select **Administration**, then **Manage**, and then **Data Forms and Ad Hoc Grids.**
2. Perform one of the following steps:
   - To open a data form folder, select a data form folder beneath **Data Form Folders**.
   - To open a data form, select a data form from the list displayed beneath **Data Form** when the appropriate data form folder is open.

   The icon next to the check box indicates the type of data form:

   - ▦ Simple data form
   - ▦ Composite data form
   - ▦ Ad hoc grid

   For information about setting up ad hoc grids, see the "Working with Ad Hoc Grids" chapter in the *Oracle Hyperion Planning User's Guide*.

Creating Simple Data Forms

To create simple data forms, define:

- Data form properties. See "Setting Data Form Grid Properties" on page 127, "Setting Dimension Properties" on page 128, and "Setting Row and Column Layout" on page 125.
- Row and column layout. See "Setting Row and Column Layout" on page 125.
- Page and POV. See "Page Axis" on page 123 and "Point of View" on page 123.
- Member selection. See Chapter 7, "Using the Member Selector."
- Display options. See "Setting Data Form Precision, Display Properties, and Other Options" on page 129.
- Business rules selection and properties. See "Using Business Rules" on page 140.
- Access permissions. See Chapter 3, "Setting Up Access Permissions."

- Formula rows and columns. See “Designing Data Forms with Formula Rows and Columns” on page 121.
- Data validation rules. See “Designing Data Forms with Data Validation” on page 122.

You can also create composite data forms, which are data forms that display several simple data forms simultaneously. See “Creating Composite Data Forms” on page 136.

➤ To create simple data forms:

1 Select **Administration**, then **Manage**, and then **Data Forms and Ad Hoc Grids.**

2 Select the folder in which to store the data form (see “Selecting and Opening Data Forms and Folders” on page 124).

3 Click **Create** above the list of data forms.

4 Select **Simple Data Form.**

5 Provide a name of up to 80 characters, and an optional description of up to 255 characters.

6 Select the **Plan Type** associated with the data form. See “Data Forms and Plan Types” on page 120.

7 **Optional**: Provide instructions for working with the data form.

8 Click **Next** to lay out the data form rows and columns. See “Setting Row and Column Layout” on page 125.

Setting Row and Column Layout

When you create data forms, the Layout tab initially contains one row and one column, and all dimensions are in POV. When creating or editing data forms, you can add rows and columns to a data form, as necessary.

Use the Layout tab to:

- Set general display properties for the data form (see “Setting Data Form Grid Properties” on page 127)
- Add data 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”)
- Select display properties for rows, columns, dimensions, and formula rows and columns (see “Setting Dimension Properties” on page 128) and “Adding Formula Rows and Columns” on page 132

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 POV.)

- Select a dimension from any axis and drag it to the destination axis to move a dimension from one axis to another.

➤ To set or update the data form layout:

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

See “Selecting and Opening Data Forms and Folders” on page 124.

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 data form Point of View. You can drag dimensions from Point of View to rows, columns, or Page. You can also drag dimensions from any area in the grid (row, column, POV, 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 **Optional: Right-click in a row or column and select Add Row or Add Column to add rows and columns to the data form.**

5 **Optional: To re-order rows or columns within a data form, right-click within a row or column, and select Move Row Up or Move Row Down for rows, or Move Column Left or Move Column Right for columns.**

6 **Select each dimension’s members.**

See Chapter 7, “Using the Member Selector.”

7 **Optional: Right-click in a row or column and select Move Dimension Up or Move Dimension Down to rearrange the order of dimensions in rows or columns.**

8 **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.**

Data form row and column properties define how rows and columns display in data forms. These properties add to row and column display specifications defined for the data form grid. See “Setting Data Form Grid Properties” on page 127.

Table 43 Data Form Row and Column Properties

Option	Description
Apply to all rows	Apply settings to all rows; available when there are two or more rows. Clear this option to set different properties for each row.
Apply to all columns	Apply settings to all columns; available when there are two or more columns. Clear this option to set different properties for each column.
Hide	Hides the column on the data form
Read-only	Creates a read-only row or column, enabling comparison of old, read-only data with new, editable data
Show separator	Creates a bold border before the segment to visually distinguish it
Suppress hierarchy	Suppresses indentation.

Option	Description
Suppress missing	Hides rows or columns without data. Clear to display rows or columns with “#MISSING” in cells when data is missing.
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 ● Use Default: Have the column width defined at the grid level
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 ● Use Default: Have the row height defined at the grid level
Validate only for users with access to this data form	If the currently logged in user does not have access to the data form, do not execute validations associated with the data form when validating the planning unit.

9 **Optional**: Add formula rows or columns. See “Adding Formula Rows and Columns” on page 132.

10 **Optional**: Add or update data validation rules. See “Including Data Validation Rules in Data Forms” on page 133.

Setting Data Form Grid Properties

Data form grid properties set the general data form row and column display.

➤ To set data form grid properties:

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

See “Selecting and Opening Data Forms and Folders” on page 124.

2 In **Grid Properties**, set general row and column properties using the information in this table:

Table 44 Data Form Grid Properties

Option	Description
Suppress missing blocks	(Rows only) Improves the performance of the **Suppress missing data** setting when suppressing a large number of rows, for example, 90% or more. The **Suppress missing blocks** setting can degrade performance if few or no rows are suppressed. Test data forms before and after using this setting to determine whether performance is improved. Also test data forms whenever you make significant changes to your application. With this setting selected, attributes may not display in data forms, certain suppressed blocks may ignore Dynamic Calc members, and row members do not display as indented.
Suppress missing data	Hides rows or columns without data. Clear to display rows or columns with “#MISSING” in cells when data is missing.

Option	Description
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

Setting Dimension Properties

You can set and edit data form dimension display properties, including whether to display the member name or alias in the data 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 data form, and then click Layout.**

See "Selecting and Opening Data Forms and Folders" on page 124.

2 **Click ▦ in a row or column to set dimension properties for rows or columns.**

3 **Select properties:**

Table 45 Dimension Properties

Property	Description
Apply to all row dimensions	Applies properties to all row dimensions.
Apply to all column dimensions	Applies properties to all column dimensions.
Apply to all page dimensions	Applies properties to all page dimensions.
Apply to all POV dimensions	Applies properties to all POV dimensions.
Member Name	Displays the member name.
Alias	Displays the member alias.
Member formula	Displays member formulas.
Hide dimension	Hides the dimension.
Show consolidation operators	Displays consolidation operators.
Start expanded	Initially displays the dimension member list expanded.

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

Setting Data Form Precision, Display Properties, and Other Options

In Other Options, you set options for data form display, functionality, and printing. The properties on this page also 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.

You can also enable account-level annotations. Users can add annotations to accounts in data 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. Note the following:

- 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 POV axis.
- Version and Scenario dimensions must be assigned to the page or POV axis.

➤ To set data form precision, display properties, and other options:

1 **Open the data form, and then click Other Options.**

See “Selecting and Opening Data Forms and Folders” on page 124.

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 46 Data Precision Examples

Value	Minimum Precision	Maximum Precision	Displayed Value
100	0	Any	100
100	3	Any number greater than or equal to 3 or None	100.000
100.12345	Any number less than or equal to 5	None	100.12345
100.12345	7	None	100.1234500
100.12345	Any number less than or equal to 3	3	100.123
100.12345	0	0	100
100.12345	2	4	100.1234
100	2	4	100.00

3 **In Smart View Option, select Enable offline usage if users can work on the data 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 Planning Properties" on page 35.

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

4 **In Display Properties, select data form options:**

- **Make data form read-only**. You cannot set this option for composite data forms.
- **Hide data form**. For example, hide data forms that are part of composite data forms or are accessed from menus or task lists.
- **Display missing values as blank**: Leave data form cells empty where data does not exist. If this option is not selected, empty cells display the text "#MISSING."
- **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 data 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 Online Help*.
- **Enable Grid Spread**

  For information about using Grid Spread, see the *Oracle Hyperion Planning User's Online Help*.
- **Enable cell-level document**: (Default) Enable users to add, edit, and view documents in cells in the data form, depending on access permissions. To prevent users from using documents in a data form, clear this option. To use cell documents, see the *Oracle Hyperion Planning User's Online Help*, Chapter 3, "Setting Up Access Permissions."
- **Message for data forms with no data**: Enter text to display in data form rows for queries without valid rows. Leave blank to display the default text, `There are no valid rows of data for this data form`.

5 **In Printing Options, set preferences for printing data form information:**

Table 47 Printing Options

Option	Description
Include supporting detail	Include supporting detail as extra rows in PDF files. Specify display format: **Normal Order**: Prints supporting detail in the same order as on the Supporting Detail page, after the member it is associated with **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
Show comments	Display text notes associated with cells
Format data	Apply number format settings from the data form to the displayed data

Option	Description
Show attribute members	If attribute members are selected in the data form, display them in PDF files
Apply precision	Apply data form precision settings (desired number of decimal points) to the displayed data in PDF files
Show currency codes	If the data form supports multiple currencies, display currency codes in the data form and in PDF files. Whether currency codes display depends on whether currency codes are present on any member in the data form. If a currency code is present on any member contained in the data form, currency codes display in the data form regardless of the selection for this check box. If currency codes are not present on members in the data form, currency codes are not displayed.
Show account annotations	If account annotations are enabled for the data form, select to display account annotations in PDF files

You can also create reports for data forms, as described in "Customizing Reports" on page 319.

6 In **Context Menus**, associate menus with the data form by selecting them from **Available Menus** and moving them to **Selected Menus**:

- ⇥ moves selections
- ⇉ moves all
- ⇤ removes selections
- ⇇ removes all

7 If you select multiple menus, click ⌃ or ⌄ to set the order in which they display.

Multiple menus display sequentially, with separators between them.

8 Select **Enable Dynamic User Variables** to allow dynamic user variables in the data form (see the *Oracle Hyperion Planning User's Online Help*).

9 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 data form, and then click **Layout.**

See "Selecting and Opening Data Forms and Folders" on page 124.

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 data form.

Adding Formula Rows and Columns

Formula rows contain formulas that apply to data form rows. Formula columns contain formulas that apply to data 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. You define a formula by selecting the formula row or column, and then enter the formula in the formula bar (see "Creating Formulas Using the Formula Bar" on page 361).

Note: Add 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 data form.

➤ To add formula rows and columns:

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

See "Selecting and Opening Data Forms and Folders" on page 124.

2 In the Layout tab, right-click **Rows** or **Columns**.

3 Select **Insert Formula Row** or **Insert Formula Column**.

4 Click **Formula Label** in the formula row or column, and then enter the formula name.

The formula bar is displayed at the top of the Layout tab.

5 Click the formula row or column to display the formula row or column properties in the right pane, and then select or clear these properties:

- **Hide** hides the row or column
- **Show separator** displays the row or column separator
- **Display formula on data form** displays the formula on the data form when you click in the row or column header.

6 For each dimension in **Formula Data Type** in the right pane, select a data type for the formula result.

The data types are:

- Currency
- Non Currency
- Percentage
- SmartList

If you select SmartList, select a Smart List from the drop-down list next to the data type.

- Date
- Text

7. **Enter the formula in the formula bar. See “Editing Formulas Using the Formula Bar” on page 361.**

8. **Optional: In the formula bar, click Select to include one of the available mathematical functions in the formula. See Appendix C, “Data Form Formula Functions.”**

9. **Click f(·)✓ to validate the formula.**

   If the formula is valid, the flag at the right of the formula bar is green. The flag is red if the formula is not valid.

10. **Optional: If there are errors in the formula, click 🚩 to display a tool tip that describes the errors. Fix the errors, and then click f(·)✓ to check your corrections.**

    When the formula is valid, the flag at the right of the formula bar is green.

11. **Click Save to save your work and continue, or Finish to save your work and close the data form.**

Including Data Validation Rules in Data 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 data 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 data forms:

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

   See “Selecting and Opening Data Forms and Folders” on page 124.

2. **In the Layout tab, right-click in the location (grid, column, row, or cell) where you want to include the data validation rule.**

3. **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.

     The menu options that display are context-sensitive, and depend on whether rules have already been added, and if you have previously selected a menu option. For example, if you right-click a cell that contains a rule and select Copy Validation Rules, the Paste Validation Rules menu option is displayed when you right-click another cell.

4. Build and validate the rules as described in “Creating and Updating Data Validation Rules” on page 169.
5. In the data form, click **Next** to continue building the data form, and then validate and save the data form.

Defining Simple Data 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 data 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 data forms, the variable name displays in Point of View. See “Managing User Variables” on page 149.

➤ To define page axis and point of view:

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

   See “Selecting and Opening Data Forms and Folders” on page 124.

2. Click and then drag that dimension to **Page** to add it to the data 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 **Properties**, select or clear options for page dimensions. See “Setting Dimension Properties” on page 128.
6. **Optional**: Click and then drag that dimension to **Point of View** to add it to the data form POV. 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. See Chapter 7, “Using the Member Selector.”
8. In **Properties**, select or clear options for point of view dimensions. See “Setting Dimension Properties” on page 128.
9. Click **Save** to save your work and continue, or click **Finish** to save your work and close the data form.

Editing Data Forms

You can edit the layout, members, and properties of both simple and composite data forms. For example, you can add formula rows or columns to a simple data form, or add data forms to a composite data form.

➤ To edit simple data forms:

1 Select the data form, then click **Edit** (see "Selecting and Opening Data Forms and Folders" on page 124).

2 Select:

a. **Properties** to edit the data form name, description and instructions. See "Creating Simple Data Forms" on page 124.

b. **Layout** to edit data form layout. See "Setting Row and Column Layout" on page 125.

c. **Other Options** to edit data form display. See "Setting Data Form Precision, Display Properties, and Other Options" on page 129.

d. **Business Rules** to change which business rules are associated with the data form, or modify business rule properties. See "Using Business Rules" on page 140.

3 Click **Finish** to save your work and close the data form.

➤ To edit composite data forms:

1 Select the data form, and then click **Edit** (see "Selecting and Opening Data Forms and Folders" on page 124).

Note: When editing a composite data form, if this message displays, "Modifications have been made to one or more included data forms; if you want to save changes to common dimensions, save the composite data form," determine what changes were made to the common dimensions of the included simple data forms before saving changes to the composite data form.

2 Select:

a. **Properties** to edit the composite data form name, description or instructions. See "Creating Composite Data Forms" on page 136.

b. **Layout** to edit data form layout and properties. See "Setting Composite Data Form Layout" on page 136.

c. **Business Rules** to change which business rules are associated with the data form, or modify business rule properties. See "Using Business Rules" on page 140.

3 Click **Finish** to save your work and close the data form.

Creating Composite Data Forms

Composite data forms display several data 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 data forms:

1 Select **Administration**, then **Manage**, and then **Data Forms and Ad Hoc Grids.**

2 Select the folder in which to store the data form. See "Selecting and Opening Data Forms and Folders" on page 124.

3 Click **Create** above the list of data forms.

4 Select **Composite Data Form.**

5 Provide a name of up to 80 characters, and an optional description of up to 255 characters.

6 **Optional**: Select **Hide Data Form** to hide the data form.

7 **Optional**: Enter instructions for the data form in **Instructions.**

8 Set the composite data form layout. See "Setting Composite Data Form Layout" on page 136.

9 Set the composite data form section properties, global layout dimensions, and common dimensions. See "Setting Composite Data Form Section Properties" on page 138.

10 Set the composite data form POV and Page display options. See "Setting Composite Data Form POV and Page Properties" on page 139

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

Setting Composite Data Form Layout

Planning provides tools that allow you to create whatever composite data form layout is best for your application. Each area in the composite data form is called a section. Initially, you specify whether to divide the composite data 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 data form layout:

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

See "Selecting and Opening Data Forms and Folders" on page 124.

2 In **Select layout**, select one option:

- C **Custom Layout** to create your own composite data form layout.

  Planning displays the selected data forms across the composite data form from left to right in the order selected.

- **2-Column Layout** to split the composite data form into two side-by-side sections divided by a vertical line.
- **2-Row Layout** to split the composite data form into two sections, one on top of the other, divided by a horizontal line.

The selected layout is displayed.

3 **Optional**: Select one or more layout options to create additional sections and control data form display. Right click in a section and select:

- **Group as Tabs** to display the data forms in that section as tabs. **Data forms per row** and **Data forms per column** are not available if you select this option.
- **Ungroup Tabs** to clear **Group as Tabs**.
- **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 data form section that contains simple data forms, the simple data 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 data forms from the split section are included in the left section, and the right section is empty.

4 **Optional**: Click ☒, and then click **OK** when prompted to remove a section from the composite data form.

Note: When you delete a section from a composite data form, the simple data forms included in that section are also deleted from the composite data form, unless they are included in other sections of the composite data form.

5 To select simple data forms for each section, drag simple data forms from the left pane to the section, or select in **Section Properties**, and then select simple data forms in the Add or Remove Data Forms dialog box.

During runtime, the simple data forms selected for the composite data form display from left to right, and then from top to bottom within each composite data form section. The display order changes if "Display data forms as Tabs" is selected. When this option is selected, the data forms display in the order selected.

Note: Composite data forms can contain only simple data forms.

Note: You can drag simple data forms between sections of a composite data form.

6 **Optional**: Use the arrow keys in **Section Properties** to rearrange data forms within a section.

7 **Optional: Click ✚ in Section Properties, and then select simple data forms from the data form selection dialog box to add or remove simple data forms from a composite data form.**

Note: You can select a simple data form from the list in **Section Properties**, and then click ✖ to remove it from the composite data form. You can also click a simple data form icon in a composite data form section, and then drag it outside the section boundaries to remove it from the composite data form.

Setting Composite Data Form Section Properties

Each section in a composite data form is associated with properties set during creation. You can edit these properties after creating a composite data form.

➤ To set composite data form properties:

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

See "Selecting and Opening Data Forms and Folders" on page 124.

2 **For each composite data form section, select the section. In Section Properties, set properties:**

Table 48 Composite Data Form Section Properties

Option	Description
Display data forms as Tabs	Displays data forms as tabs in Layout, in Preview mode, and at runtime.
Section Name	Displays the section name at the top of the section in Preview mode and at runtime.
Section Height	Select: ● **Automatic** to have Planning set the height. ● Percentage sign to set section height to a percentage of the composite data form height.
Section Width	Select: ● **Automatic** to have Planning set the width. ● Percentage sign to set section width to a percentage of the composite data form width.
Data forms per row	Select one: ● **Automatic** to have Planning set the number. ● Select a number from 1 to 20. The default is one data form per row. If Data forms per column is set to a value other than Automatic, Data forms per row is set to Automatic.

Option	Description
Data forms per column	Select one: ● **Automatic** to have Planning set the number. ● Select a number from 1 to 20. The default is one data form per column. If Data forms per row is set to a value other than Automatic, Data forms per column is set to Automatic.

3 **Optional**: Specify a label for the tab of simple data forms displayed as tabs in the composite data form:

a. Select the data form in **Section Properties**.

b. Click ✏ to add a label. Enter the label, and then click within the Section Properties window or press **Tab**.

Note: When you open a composite data form, the tool tip displays the data form name, not the data form label, for a simple data form within a composite data form.

Setting Composite Data Form POV and Page Properties

The composite data form point of view and Page dimension properties specify where within a composite data form each POV and Page dimension name displays. When you select a section in a composite data form, the right panel displays:

- **Global Layout Dimensions**, which lists the POV and Page dimensions that display in the composite data form heading.
- **Common Dimensions**, which lists the POV and Page dimensions common to all the simple data forms included in the selected composite data form section.

You specify where common dimensions display in composite data forms. Common dimension display choices are:

- **Local**: Displays the dimension name in the simple data form heading
- **Section**: Displays the dimension name in the section heading

  Note: Only dimensions that are common to all simple data 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 data form heading

  Note: Only dimensions that are common to all simple data forms in all sections of the composite data form and that contain the same members can be designated as Global.

➤ To set composite data form POV and page options:

1 Take an action:

- To update the current data form, click **Layout**.
- To open a data form for editing, click **Edit**, and then click **Layout**.

2. In **Common Dimensions**, for **POV dimensions** and **Page dimensions**, specify where the dimension name displays within the composite data form. Select:

   - **Local** to display in the simple data form heading
   - **Section** to display in the section heading
   - **Global** to display in the composite data form heading

3. **Optional**: Click ✏ to the right of a common dimension, and then enter a label for that dimension.

   Click ⌫ to remove the dimension label.

Using Business Rules

You can create business rules using Calculation Manager. 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 data 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 with Business Rules, see the *Hyperion Business Rules Administrator's Guide*.
- Creating and updating business rules with Calculation Manager, see the *Oracle Hyperion Calculation Manager Designer's Guide*.
- Using runtime prompts, see "About Runtime Prompts" on page 142.

Selecting Business Rules

On the Business Rules tab, you can associate multiple business rules with a data form, by plan type. Users can launch associated business rules from the data form to calculate and allocate values. You can set whether each business rule associated with a data form automatically launches when the data form is opened or saved.

➤ To select business rules for data forms:

1. Take an action:

   - To update the current data form, click **Business Rules**.
   - To open a data 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 data form, and move them to **Selected Business Rules**. See Chapter 7, "Using the Member Selector."

By default, the Calculate Data Form and Calculate Currencies business rules are selected. Calculate Data Form is automatically created for data forms to calculate subtotals. Calculate Currencies is created for data 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 Data Form to prevent planners from calculating data in data 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 141.

6 Click **Save** to save your work and continue creating or editing the data form, or click **Finish** to save your work and close the data form.

Setting Business Rule Properties

You can specify whether business rules associated with data forms launch automatically when users open or save the data 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 POV axes.

➤ To set business rule properties:

1 Take an action:

- To update the current data form, click **Business Rules**.
- To open a data form for editing, click **Edit**, and then click **Business Rules**.

2 Click **Properties** in the **Business Rules** tab.

3 Select **Run on Load** next to a business rule to launch it automatically when the data form is opened.

Business rules having runtime prompts cannot launch on load.

4 Select **Run on Save** next to a business rule to launch it automatically when saving a data form.

If business set to **Run on Save** contain runtime prompts, users are prompted to enter the runtime prompt value before the data form is saved.

5 **Optional**: If a business rule has runtime prompts, select **Use Members on Data Form** to match the default member selection on the runtime prompt window to the current members in the page and POV axes of the open data form.

To learn how this option interacts with other settings and conditions, see "Understanding Runtime Prompts" on page 142.

6 **Optional: To hide the runtime prompt value from the user, select Hide Prompt, which automatically selects Use Members on Data Form.**

After saving the data form, the next time you return to this page, **Use Members on Data 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 data form's Page/POV)
- No dimensions are repeated in the runtime prompt

7 Click **OK.**

About Runtime Prompts

When launched, business rules can prompt users for such variables as members, text, dates, or numbers. Prompts should be specific and tell users what type of data is expected. For example:

- Select a month.
- Enter the expected number of customer visits per quarter.
- What percentage change in earnings do you expect next month?

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 141. To understand how other settings and conditions affect runtime prompts, see "Understanding Runtime Prompts" on page 142.

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 Planning Properties" on page 35.

Understanding Runtime Prompts

The display and values of runtime prompts are affected by such aspects as:

- Whether there are valid members on the data form's Page/POV and whether the **Use Members on Data Form** and **Hide Prompt** options on the **Business Rule Properties** tab are selected (see "Setting Business Rule Properties" on page 141)
- Whether the **Runtime Prompt** option is set during form design or set when designing the runtime prompt (for business rules created with Business Rules, see the *Hyperion Business*

Rules Administrator's Guide and for business rules created with Calculation Manager, see the *Oracle Hyperion Calculation Manager Designer's Guide*)

- Whether the Use Last Value property is set when designing the business rule

Principles:

1. When launched from a data form, the values of members in the Page/POV take precedence if the **Use Members on Data 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/POV members.

   This is not the case when business rules associated with composite data 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 Data Form** setting is ignored, hidden runtime prompts get design-time values, and the last saved value takes precedence.

2. 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.

3. The **Use Members on Data 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).

4. 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 Page/POV. The runtime prompt is displayed with some values pre-filled from the Page/POV and others filled according to Principles 1 and 2.

This table describes the result on runtime prompts of these settings and conditions:

Table 49 How Member Availability and Other Settings Affect Runtime Prompts

Availability of member on the Page/POV	Use Members on Data Form option is selected	Hide Runtime Prompt property is set during runtime prompt design	Hide Prompt option is selected for the data form	Result on Runtime Prompt
The member is available on the Page/POV to use as the runtime prompt value.	Yes	Yes	Yes or No Setting is ignored	The business rule runs without displaying the runtime prompt to users. Instead, the runtime prompt value is taken from the Page/POV member.
The member is available on the Page/POV to use as the runtime prompt value.	Yes	No	Yes	If all runtime prompts can be pre-filled from the Page/POV 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 Page/POV context, then all runtime prompts display, with values pre-filled wherever possible. All others follow Principle 2.
The member is available on the Page/POV to use as the runtime prompt value.	Yes	No	No	The runtime prompt is displayed to users, with values pre-filled from the Page/POV.
The member is not available on the Page/POV to use as the runtime prompt value.	Yes	Yes	Yes or No Setting is ignored	The business rule displays the runtime prompt to users, with values pre-filled according to Principle 2. For example, the data form context cannot be passed because the dimension of the runtime prompt is on rows or columns, so the Hide Prompt setting is ignored and the runtime prompt displayed.
The member is not available on the Page/POV to use as the runtime prompt value.	Yes	No	Yes	The runtime prompt is displayed to users, with values pre-filled according to Principle 2.
The member is not available on the Page/POV to use as the runtime prompt value.	Yes	No	No	The runtime prompt is displayed to users, with values pre-filled according to Principle 2.

Availability of member on the Page/POV	Use Members on Data Form option is selected	Hide Runtime Prompt property is set during runtime prompt design	Hide Prompt option is selected for the data form	Result on Runtime Prompt
The member is available on the Page/POV to use as the runtime prompt value.	No	Yes	Not available	The business rule runs without displaying the runtime prompt to users. Instead, the design-time values are used.
The member is available on the Page/POV to use as the runtime prompt value.	No	No	Not available	The runtime prompt is displayed to users, with values pre-filled according to Principle 2.
The member is not available on the Page/POV to use as the runtime prompt value.	No	Yes	Not available	The business rule runs without displaying the runtime prompt to users. Instead, the design-time values are used.
The member is not available on the Page/POV to use as the runtime prompt value.	No	No	Not available	The runtime prompt is displayed to users, with values pre-filled according to Principle 2.

When hidden runtime prompt values are ambiguous, note:

- If the data 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 context, the runtime prompt displays with values pre-filled context values and design time values for dimensions on the row or column. For example, if the Cross Dimension has runtime prompts for Period, Entity, and Scenario, and Entity is defined on the row, then the runtime prompt displays with the POV Scenario -> design time value for Entity -> Page Period.
- If the 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 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 or Business Rules sequences
- 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. In Calculation Manager, a variable cannot be hidden unless a value is provided. 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.

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 Online Help*).
   - Creating a runtime prompt `ASCII` file that contains, on separate lines, each runtime prompt name and its value, separated by a double colon (::). For example:

     ```
     CopyDataFrom::Jan
     ```

     ```
     CopyDataTo::Apr
     ```

     This file specifies that the value for the runtime prompt named `CopyDataFrom` is `Jan`, and that the value for the runtime prompt named `CopyDataTo` is `Apr`.

     Save the file in the `planning1` directory (for the full path to `planning1`, see “About EPM Oracle Instance” on page 45). Alternately, when executing the utility, specify the full path to the runtime prompt file.

2 **From the `planning1` directory, enter this command at the Command Prompt, one space, and the parameters, each separated by a space:**

```
CalcMgrCmdLineLauncher.cmd [-f:passwordFile] /A:appname /U:username /
D:database [/R:business rule name| /S:business ruleset name] /
F:runtime prompts file [/validate]
```

Table 50 CalcMgrCmdLineLauncher Parameters

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 44.	No
`/A:appname`	Specify the Planning application from which the business rule is launched	Yes
`/U:username`	Specify the administrator's user name	Yes
`/D:database`	Specify the name of the plan type against which the calculation is launched	Yes
`[/R:business rule name` | `/S:business ruleset name]`	Specify the name of the business rule or ruleset to be launched	Yes. Specify either a business rule or a ruleset, but not both.
`/F:runtime prompts file`	Specify the name of the file containing business rule's runtime prompt names and values	Yes, when launching a business rule having a runtime prompt. 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.
`[/validate]`	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.	No
`/?`	Print the syntax and options for `CalcMgrCmdLineLauncher.cmd`	No

For example to launch the rule named Depreciate, using runtime prompt values in a file called `Values.txt`, enter:

```
CalcMgrCmdLineLauncher.cmd /A:planapp /U:admin /D:plan1 /
R:Depreciate /F:Values.xml
```

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")0, ERROR)
    ```

  For instructions on customizing the `HspCustomMsgs` file, see “Customizing Text, Color, and Images” on page 326.
- You cannot use the function in member formulas.

For more information on the `@RETURN` function, see the *Oracle Essbase Technical Reference*.

Previewing Data Forms

While you are designing data forms, you can preview the dimensions that are assigned to the POV, columns, rows, and page axes. Previewing displays member attributes, alias, and data associated with data forms, although new data cannot be entered.

Previewing completes regular data form design validation checks, and checks for proper evaluation of any data validation rules included in the data form. Data validation rules must be properly completed before the data form can be saved. In addition, data validation rules are saved as part of the data form. If you do not save changes to a data form, any data validation rule changes made after the data form was last saved are lost.

➤ To preview a data form's design:

1. With a data form open, click **Preview.**
2. Resolve any issues reported during the design validation checks, including any issues with data validation rules.
3. Save the data form to ensure that updates are saved, including any changes to data validation rules.

Managing User Variables

You can set user variables to limit the number of members displayed on a data 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 data form. See "Defining Simple Data Form Page and Point of View" on page 134.

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 "Selecting User Variables" on page 149.
3. When designing the data form, associate the user variable with the data form.

   See "About Selecting User Variables as Members" on page 166.
4. Instruct users to select a member for the user variable associated with the data form.

   Before users can open data 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 data form or in preferences. See the *Oracle Hyperion Planning User's Online Help*.

Selecting User Variables

➤ To select user variables:

1. Select **Administration**, then **Manage**, then **User Variables.**
2. Click **Add.**
3. On the **Add User Variable** page, for **Dimension Name**, select the dimension for which to create a user variable.
4. For **User Variable Name**, enter the name of the user variable.
5. **Optional**: Select **Use Context** to allow user variables to be used in the POV. With this setting, the value of the user variable changes dynamically based on the context of the data form.
6. Click **OK.**

You can now associate the user variable with a data form. See "About Selecting User Variables as Members" on page 166. After that, planners can select members for the user variable. See the *Oracle Hyperion Planning User's Online Help*.

Deleting User Variables

➤ To delete user variables:

1 Select **Administration**, then **Manage**, then **User Variables.**

2 Select the user variable to delete.

3 Click **Delete.**

4 Click **OK.**

Printing Data Form Definitions

Administrators can print data form definition reports that include information on dimension members, business rules, access permissions, and other data form components. You can also create reports for data form definitions, as described in "Customizing Reports" on page 319.

➤ To create and print data form definition reports:

1 Select **Administration**, then **Application**, then **Reports.**

2 Select **Data Forms.**

3 Select the data form definitions to print:

- To select data forms, move them to **Selected Data Forms** by selecting them and clicking → or ⇉.
- To remove data forms, select them from **Selected Data Forms** and click ← or ⇇.

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
- Data form access permissions
- Associated business rules

7 **To print the report, click 🖨 on the Adobe toolbar.**

Note: To ensure that multibyte characters display in reports, see “Multibyte Characters” in the *Oracle Hyperion 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`.

Searching for Data Forms

➤ To search for data forms:

1 **Select Administration, then Manage, then Data Forms and Ad Hoc Grids.**

2 **For Search, enter part or all the data form name.**

Ignoring capitalization, Search finds the next match.

3 **Click to search forward (down) or to search backwards (up).**

Moving Data Forms and Ad Hoc Grids

➤ To move data forms:

1 **Select the data form.**

See “Selecting and Opening Data Forms and Folders” on page 124.

You can move multiple data forms simultaneously if they are in the same folder.

2 **Click Move.**

3 **Select the destination folder.**

4 **Click OK.**

Deleting Data Forms

➤ To delete data forms:

1 **Select the data form.**

See “Selecting and Opening Data Forms and Folders” on page 124.

2 **Click Delete.**

3 **Click OK.**

Renaming Data Forms

➤ To rename data forms:

1. **Select the data form.**

   See "Selecting and Opening Data Forms and Folders" on page 124.

2. **Click Rename.**

3. **Enter the new name and click OK.**

Overview of Importing Data Form Definitions

Use the `ImportFormDefinition` utility to import a data form definition, but not data, from a text-based file into a Planning data 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 data form definitions:

1. Prepare the data form for importing.

   See "Preparing Data Forms" on page 152.

2. Prepare the data file.

   See "Preparing Data Files" on page 152.

3. Run the utility.

   See "Importing Data Form Definitions" on page 153 and "ImportFormDefinition Examples" on page 154.

Preparing Data Forms

Before importing rows and columns for data form definitions, create the data form by setting up dimensions on the row, column, page and POV, depending on your requirements. Typically you define column layout when defining data forms and use `ImportFormDefinition` to import only rows. The resulting data form looks like the data form definition.

The rows imported from the data file are based on members specified on the data form and import options, determining which data is imported. See "ImportFormDefinition Examples" on page 154.

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 data form, not the format of the data file, determines how the resulting data form displays. You can use the same data file to load different data forms.

Importing Data Form Definitions

The `ImportFormDefinition` utility is in the `planning1` directory. For the full path to `planning1`, see “About EPM Oracle Instance” on page 45.

➤ To launch `ImportFormDefinition`:

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

```
ImportFormDefinition [-f:passwordFile] /A:appname /U:username /
F:formname /D:filename [/AR] [/AC] [/SR] [/SC] [/KC] [/KR]
```

Table 51 ImportFormDefinition Utility Parameters

Setting	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 44.	No
`/A`	Application name.	Yes
`/U`	Administrator user name.	Yes
`/F`	Data form name.	Yes
`/D`	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.	Yes
`/AR`	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.	No
`/AC`	Add columns from the data file (on by default). Disable by specifying `/-AC`.	No

Setting	Purpose	Required?
/KC	Keep member selections for columns in the data form (on by default). Clear columns by specifying /-KC. Data forms must have at least one column definition. If you clear columns but do not add them, the data form is not saved, and an error displays.	No
/KR	Keep member selections for rows in the data form (on by default). Clear member selections in rows by specifying /-KR. Data forms must have at least one row definition. If you clear rows but do not add them, the data form is not saved, and an error displays.	No
/SR	Sort rows in the data form (on by default). Disable by specifying /-SR.	No
/SC	Sort columns in the data form (on by default). Disable by specifying /-SC.	No

2 **If prompted, enter your password.**

For example:

```
ImportFormDefinition /A:MyPlan /U:Admin /F: "My Budget" /
D:exportfilename /AR /-AC
```

Creating the data form can take some time, depending on the amount of data in the file.

`ImportFormDefinition` imports the definition to the Planning data form, ensuring that all cells having data in the data file are represented on the data form. For cells in the data file that contain “#MISSING,” rows or columns are not added to the data form.

Notes:

- If you disable importing rows or columns, `ImportFormDefinition` filters imported data by rows or columns defined on the data form.
- If you run `ImportFormDefinition` more than once, it merges new results with the existing data 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 data 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 data 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 data form design.

   For example, to load accounts on the row, place the Account dimension root on the data 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 data form columns, pages, and POV. 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 data form. Rows are filtered by the members on the POV. If the data file contains data for the salary account for 2008, but only 2009 is on the POV, the salary account is not added to the row although it exists in the data file.

Importing and Exporting Data Forms

Administrators can use `FormDefUtil.cmd` (Windows) or `FormDefUtil.sh` (UNIX) to move data form definitions between Planning applications. You can export or import data 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 45.

Note: XML files for composite data 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 data 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|exportfilename|formname|-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 44.	No
`import|export`	Import or export the data form definition.	Yes
`filename|formname|-all`	When used with `import`, specify the XML file containing the data form definition. When used with `export`, specify the data form to export to XML. Use `-all` with `import` or `export` to import or export all XML files or data form 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 data form definitions to export. When used with `import`, the name of the Planning application to which to import the data form definition.	Yes

2 If prompted, enter your password.

When you export data 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 45.) You can copy the utility to any directory and launch it from there to save files to another directory.

Examples:

- To import one file:

  ```
  FormDefUtil.cmd import c:\EPM_ORACLE_INSTANCE\Planning
  \planning1\form1.xml localhost admin APP1
  ```

- To export one file:

  ```
  FormDefUtil.cmd export Form1 localhost admin APP1
  ```

- To export all data form definitions:

  ```
  FormDefUtil.cmd export -all localhost admin APP1
  ```

- To import all data form definitions:

  ```
  FormDefUtil.cmd import -all localhost admin APP1
  ```

Planning Offline Considerations

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

- For information about installing and configuring offline Planning, see *Oracle Hyperion Enterprise Performance Management System Installation and Configuration Guide*.
- For information about using the Install Smart View menu item available from the Tools menu, see *Oracle Hyperion Enterprise Performance Management System Workspace User's Guide* or online help.
- For a user to take a data form offline, the administrator must assign that user the Offline User role. For information about the Offline User role, see *Oracle Hyperion Enterprise Performance Management System Security Administration Guide* or online help.

Offline Data Form Considerations

- Data forms used offline should be self-contained.

  To ensure that values calculated offline are correct, all necessary dependencies pertinent to data forms taken offline must be available offline. All members, data forms, member formulas, business rules, and dynamic calculations on which a data 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 data 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 data form member is deleted on the server while a user is working with that data 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 data form property Enable Offline Usage set when a data 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 data form simultaneously. An offline user can lose access to members taken offline if an online user working with the same data form clears the values in a row or column that has Suppress Missing Data set.
- Smart View does not currently support hidden data forms; they are not downloaded when users take data forms offline that are associated with hidden forms.
- Smart View does not currently support composite data forms; they are not listed when users select data 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.

7 Using the Member Selector

In This Chapter

Selecting Members

Use the Member Selection dialog box to select members and other information for Planning features. For example, you can select members for data 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 162.

   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**, **Descendents (inc)**, **Ancestors**, **Ancestors (inc)**, **Siblings**, **Siblings (inc)**, **Parents**, **Parents (inc)**, **Children**, **Children (inc)**, or **Level 0 Descendants**.
- Keep only by Attributes. If attributes are defined, select options for **Attributes**, **Operator**, and **Value**, and then click **OK**. Multiple attributes can be selected.
- Keep only by levels or generations. Select **Level** or **Generation**, select the levels or generations to display, and then click **OK**.
- Refresh the display to remove filters.

4 From the members list, make a selection.

Only members to which you have access are displayed.

Notes:

- **For runtime prompts only**: The displayed hierarchy is for the application and plan type against which the business rule is launched. For runtime prompts, members are displayed that satisfy runtime prompt limits.
- **For business rules or validation rules having Cross Dimension or Member Range runtime prompts only**: From the **Select Dimension** list, select a dimension that the business rule designer set for this runtime prompt.
- **For shared members only**: Shared members are displayed in this format: `*shared_member*.*parent_member*(shared)`. When they are selected on the right side of the dialog box, only the shared member name is displayed.

5 **Optional**: Click ⊟ or ⊞ to collapse or expand the hierarchy.

6 Move members to or from the **Selected Members** list:

- Click 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 52 Member Relationships

Relationship	**Members Included**
Member	The selected member
Descendants	All descendants of the selected member, excluding the selected member
Descendants (inc)	The selected member and its descendants
Ancestors	All members above the selected member, excluding the selected member
Ancestors (inc)	The selected member and its ancestors
Siblings	All members from the same level in the hierarchy as the selected member, excluding the selected member
Siblings (inc)	The selected member and its siblings
Parents	The member in the level above the selected member
Parents (inc)	The selected member and its parent
Children	All members in the level immediately below the selected member
Children (inc)	The selected member and its children
Level 0 Descendants	All descendants of the selected member that have no children

Optional: Click the Function Selector icon on the right side of the dialog box to insert functions for the members selected in the right pane.

7 If variables or attributes are defined, click the **Variables** tab to select members for user variables, substitution variables, and attributes. Members of each category are displayed as children. Only members to which the user has read access are displayed in data forms.

- **User Variables**: Select members as described in "About Selecting Substitution Variables as Members" on page 165 and "Managing User Variables" on page 149.
- **Substitution Variables**: Select members as described in "About Selecting Substitution Variables as Members" on page 165. The selection depends on the preference setting for **Enable the Display of Substitution Variables**, described in "Specifying System Settings" on page 231.
- **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 162.

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

Table 53 Attribute Value Selection

Operator	**Attribute Values Included**
Equal	Equal to the selected attribute

Operator	Attribute Values Included
NotEqual	Not equal to the selected attribute
Greater	Greater than the selected attribute
GreaterOrEqual	Greater than or equal to the selected attribute
Less	Less than the selected attribute
LessOrEqual	Less than or equal to the selected attribute

8 **Be sure the member selection options are appropriate for the context from which you invoked the member selector dialog box.**

9 **Click OK.**

Using Wildcards in Searches

You can use these wildcard characters to search for members.

Table 54 Wildcard Characters

Wildcard	Description
?	Match any single character
*	Match zero or multiple characters. For example, enter “sale*” to find “Sales” and “Sale” because * includes zero or more characters after the word “sale.” The default search uses the * wildcard. For example, entering “cash” searches for “*cash*” and returns “Restricted Cash”, “Cash Equivalents”, “Cash”, and “Noncash Expenses” because the word “cash” appears within each matched item.
#	Match any single number (0-9)
[list]	Match any single character within a specified list of characters. You can list specific characters to use as wildcard. For example, enter [plan] to use all the letters within the brackets as a single wildcard character. You can use the “-” character to specify a range, such as [A-Z] or [!0-9]. To use the “-” character as part of the list, enter it at the beginning of the list. For example, [-@&] uses the characters within the brackets as wildcard characters.
[!list]	Match any single character not found within a specified list of characters. The “-” character can also be used to indicate a range, such as [!A-Z] or [!0-9].

About Selecting Attribute Values as Members

If attribute members are defined, you can select attribute values on the Member Selection page. For attribute members, selecting a non-level 0 attribute selects all level 0 descendants and applies the operator to each. For attributes of type numeric, date, and Boolean (where false = 0 and true = 1), evaluation is based on the minimum and maximum values. For text attributes, evaluation is based on the position from top to bottom in the hierarchy. The top position has the lowest value, and the bottom position has the highest value.

Example: Numeric attribute

In this example, the selected operator is applied to each level 0 descendant, based on the numeric value. For example, selecting NotEqual and Small in the Member Selection page includes all values not equal to 1 and not equal to 2, so the selection includes 3, 4, 5, and 6. Selecting Greater and Small includes all values greater than 1 or greater than 2, so the selection includes 2, 3, 4, 5, and 6.

```
Size

Small

1

2

Medium

3

4

Large

5

6
```

Table 55 Example: Numeric Attribute Evaluation

Selected Operator	Selected Attribute Value	Result	Explanation
Equal	Large	5, 6	The Equal operator is applied to all level 0 descendants of Large, which includes 5 and 6.
Less	Medium	1, 2, 3	The Less operator is applied to all level 0 descendants of Medium. This includes values < 3 OR < 4, which results in 1, 2, and 3.
Greater	Medium	4, 5, 6	The Greater operator is applied to all level 0 descendants of Medium. This includes values > 3 OR > 4, which results in 4, 5, and 6.
GreaterOrEqual	Medium	3, 4, 5, 6	The GreaterOrEqual operator is applied to all level 0 descendants of Medium. This includes values >=3 OR >= 4, which results in 3, 4, 5, and 6.
LessOrEqual	Medium	1, 2, 3, 4	The LessOrEqual operator is applied to all level 0 descendants of Medium. This includes values <=3 OR <=4, which results in 1, 2, 3, and 4.
NotEqual	Medium	1, 2, 5, 6	The NotEqual operator is applied to all level 0 descendants of Medium. This includes values not equal to 3 AND not equal to 4, which results in 1, 2, 5, and 6.

Example: Text attribute

For text attributes, the selected operator is applied to each level 0 descendant based on its position in the hierarchy, from top (lowest value) to bottom (highest value).

In this example, Envelope is at the top position and has the lowest value. Packet has the next higher value, followed by Box, Carton, Barrel and Crate. Crate is at the bottom position and has the highest value.

For this text attribute, selecting Less and Small includes values that are less than Envelope or less than Packet. Because Envelope is less than Packet, the resulting selection includes only Envelope. Likewise, selecting Greater and Large includes values that are greater than Barrel or greater than Crate, so the resulting selection includes only Crate.

```
Containers

Small

Envelope

Packet

Medium

Box

Carton

Large

Barrel

Crate
```

Table 56 Example: Text Attribute Evaluation

Selected Operator	Selected Attribute Value	Result	Explanation
Equal	Medium	Box, Carton	The Equal operator is applied to all level 0 descendants of Medium, which includes Box and Carton.
NotEqual	Medium	Envelope, Packet, Barrel, Crate	The NotEqual operator is applied to all level 0 descendants of Medium. This includes values not equal to Box AND not equal to Carton, which results in Envelope, Packet, Barrel, and Crate.
Less	Medium	Box, Packet, Envelope	The Less operator is applied to all level 0 descendants of Medium. This includes everything at a lower position than Carton OR a lower position than Box, which results in Box, Packet, and Envelope.
LessOrEqual	Medium	Envelope, Packet, Box, Carton	The LessOrEqual operator is applied to all level 0 descendants of Medium. This includes everything at the same position as Carton OR at a lower position than Carton, which results in Envelope, Packet, Box, and Carton.

About Selecting Members for Data Forms

When selecting members for data 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 50.

- The order of members in the Selected Members list determines the order on data forms. To change the order, select a member and click the Up or Down Arrow above the selected members list.
- In the Layout tab of the Data 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 131.
- For data 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 data form. For example, for a data form that contains members Acct1, Acct2, and Acct3 in Column A, if you select these members with **Place Selection in Separate Columns** selected, Acc1 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 data 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 131.
- To set display, functionality, and printing options, see "Setting Data Form Precision, Display Properties, and Other Options" on page 129.
- To set up reports that include data forms or data form definitions, see "Customizing Reports" on page 319.

About Selecting Substitution Variables as Members

Substitution variables act as global placeholders for information that changes regularly. Each variable has an assigned value that can be changed centrally on the Essbase server. Substitution variables are especially useful for developing and reporting on rolling forecasts. When you select substitution variables as members on the data 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 data form or the report script.

Notes:

- When you open or calculate values on data forms, Essbase 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` property and setting its value in seconds.
- You create and assign values to substitution variables using Administration Services Console or ESSCMD. These substitution variables are then available in Planning when you select members for a data form.
- Substitution variables must be appropriate for the context in data 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 on the Essbase server, application, or database 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 data form is opened). It does not check when you design data forms, so you should test substitution variables by saving and opening data 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 data form. For information about logs, see the *Oracle Hyperion Enterprise Performance Management System Installation and Configuration Troubleshooting Guide*.

➤ To specify substitution variables in data forms:

1. Create the data form (see "Creating Simple Data Forms" on page 124).
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 data forms, enabling planners to focus only on certain members, such as a department. Before you can associate a user variable with a data form, you must create the user variable. See "Managing User Variables" on page 149.

When you create data forms with user variables, planners must select values for the variable before opening data forms. For example, if you create a user variable called Division, planners must select a division before working in the data form. The first time planners select a variable for a data form, they do so in preferences. After that, they can update the variable in preferences or in the data form.

➤ To select user variables for data forms:

1 Create the data form (see "Creating Simple Data Forms" on page 124).

2 On **Member Selection**, select user variables the same way you select members, using the arrows to move substitution variables to and from **Selected Members.**

  User variables are displayed for the current dimension. For example, user variables for the Entity dimension might display as follows:

  ```
  Division = [User Variable]
  ```

  When selected, a user variable is preceded by an ampersand. For example:

  ```
  Idescendants(&Division)
  ```

3 Click **OK.**

8 Managing Data Validation

In This Chapter

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 data 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 184.

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 **Data Forms and Ad Hoc Grids.** Create or edit a data form, and then click the Layout tab in the Data 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.

When you hover the cursor over cells in the Layout tab, a validation message is displayed if the cell contains a validation rule. For example, if a cell contains one or more data validation rules for the column, this message is displayed: `Column-level validation rule(s) exist`.

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 173.

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 174.

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 174.

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 183.

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 174.
- 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 172.

7 **When you are ready to enable the rule to make it available in the data 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 data 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 Data Form Management page, view and update rules.**

- In the **Validation Rules** pane on the right side of the Data Form Management page, add, edit, or delete rules by clicking ✚, ✎, or ✖.
- To view rules at the current level or higher, click in the Manage Data 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 173.
- To prevent validations associated with the data form from executing when validating the planning unit if the currently logged in user does not have access to the data form, select **Validate only for users with access to this data form**.

10 **In the Data Form Management page, preview and validate the data form, resolve any validation errors, and then save changes.**

For data forms that have data validation rules enabled, rules are validated when the data form is loaded or saved. Data validation rules are saved when the data form is saved. See “Creating Simple Data Forms” on page 124.

When users open the data form, they can view and resolve validation messages using the Data Validation Messages pane. See the *Oracle Hyperion Planning User’s Online Help*.

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 data 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 169.

2 **In the Process Cell dialog box, set how the cell should appear in data 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 as a tooltip in the data 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 the *Oracle Hyperion Planning User’s Online Help*.
- 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 208.

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 data form.

2 Select the rule name, and then double-click the rule or click **OK** to view details.

Order of Evaluation and Execution for Data Validation Rules

For data validation rules in data 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 57 Priority for Rules in Data Forms

Default Condition Priority	Validation Message	Color	Do Not Promote
1 (lowest)	X		
1		X	

Default Condition Priority	Validation Message	Color	Do Not Promote
1	X	X	
2			X
3	X		X
4		X	X
5 (highest)	X	X	X

Conditions Supported by the Rule Builder

These conditions are supported by the data validation rule builder: If, Else, Else If, Then, Check Range, and Range.

For details and examples of the values supported by these conditions, see these sections:

- **If, Else, Else If**: "If Condition Values" on page 174.
- **Then**: "Then Condition Values" on page 182.
- **Check Range, Range**: "Range Condition Values" on page 182.

If Condition Values

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

- "Current Cell Value" on page 175
- "Cell Value" on page 175
- "Column Value" on page 176
- "Row Value" on page 177
- "Cross Dim Member" on page 178
- "Member Name" on page 179
- "Member" on page 179
- "Account Type" on page 180
- "Version Type" on page 180
- "Variance Reporting Type" on page 180
- "UDA" on page 181
- "Attribute" on page 181

For information on other conditions, see "Conditions Supported by the Rule Builder" on page 174.

Current Cell Value

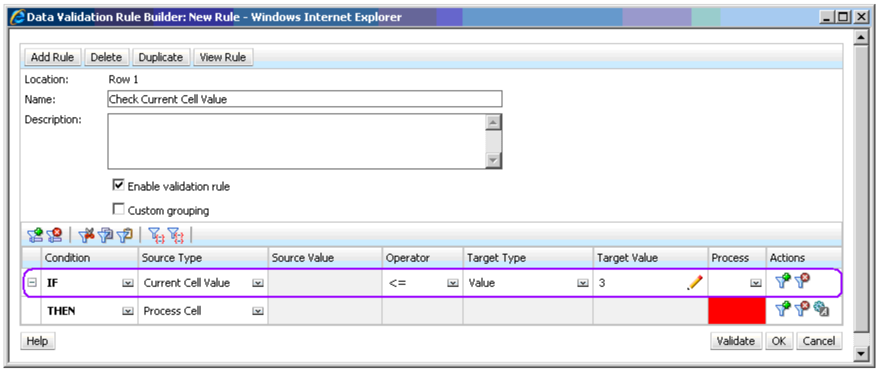
Action:

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:


Data Validation Rule Builder: New Rule - Windows Internet Explorer
Add Rule
Delete
Duplicate
View Rule
Help
Validate
OK
Cancel

Location: Row 1

Name: Check Current Cell Value

Description:

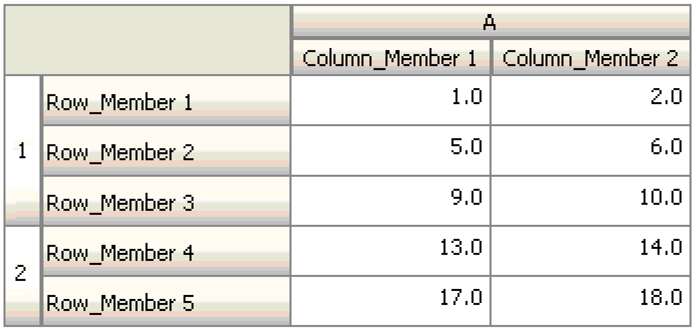
☑ Enable validation rule

□ Custom grouping

Condition	Source Type	Source Value	Operator	Target Type	Target Value	Process	Actions
IF	Current Cell Value		<=	Value	3		
THEN	Process Cell					🟥	

Condition Evaluation:

With the condition shown in the previous figure, the cells in Row 1 with member Row_Member1 will turn red when the condition is evaluated.



		A	
		Column_Member 1	Column_Member 2
1	Row_Member 1	1.0	2.0
	Row_Member 2	5.0	6.0
	Row_Member 3	9.0	10.0
2	Row_Member 4	13.0	14.0
	Row_Member 5	17.0	18.0

Cell Value

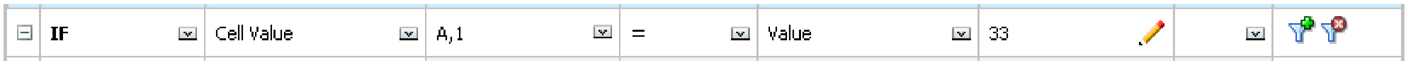
Action:

The action is performed 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.

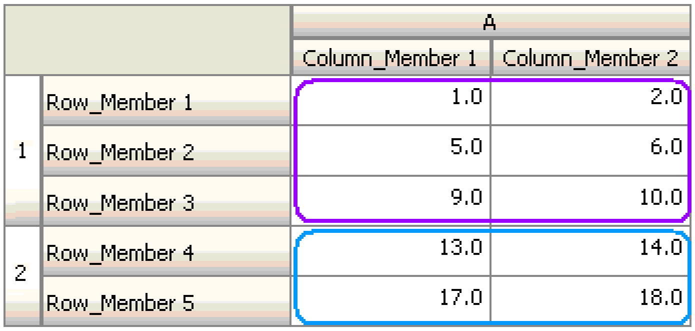
Condition Definition:



IF	Cell Value	A,1	=	Value	33

Condition Evaluation:

A design-time cell can expand to one or more data cells at data entry time, as shown in the following figure. The value for the cell is the sum of values in all the expanded data cells. For example, the value for cell A1 is the sum of the values in the cells outlined in purple (1+2+5+6+9+10=33), and the value for cell A2 is the sum of the values in the cells outlined in blue (13+14+17+18=62).



		A	
		Column_Member 1	Column_Member 2
1	Row_Member 1	1.0	2.0
	Row_Member 2	5.0	6.0
	Row_Member 3	9.0	10.0
2	Row_Member 4	13.0	14.0
	Row_Member 5	17.0	18.0

Column Value

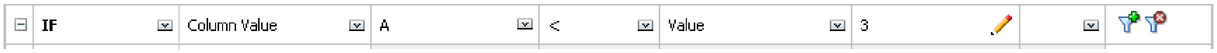
Action:

The action is performed 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:

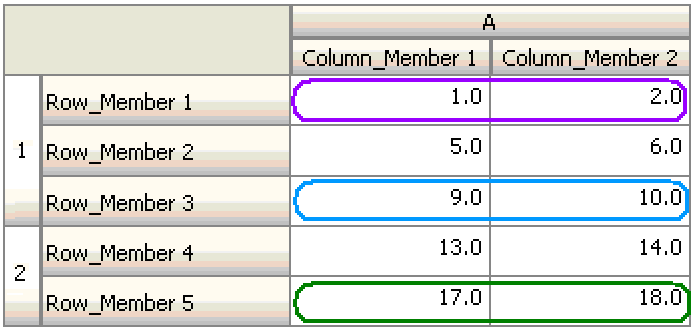


IF	Column Value	A	<	Value	3

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.



		A	
		Column_Member 1	Column_Member 2
1	Row_Member 1	1.0	2.0
	Row_Member 2	5.0	6.0
	Row_Member 3	9.0	10.0
2	Row_Member 4	13.0	14.0
	Row_Member 5	17.0	18.0

Row Value

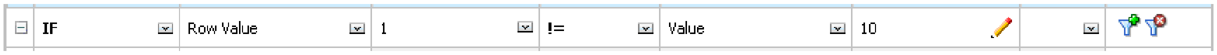
Action:

The action is performed 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:

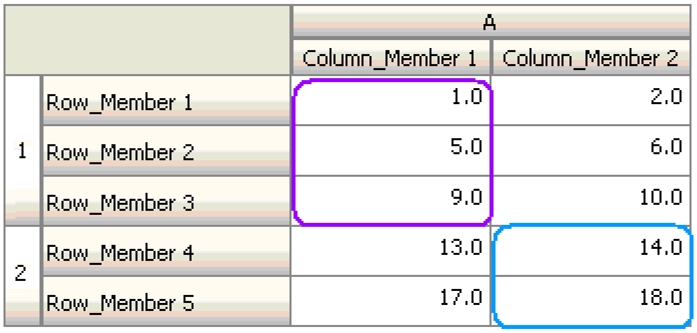


IF	Row Value	1	!=	Value	10

Condition Evaluation:

A design time row can expand to one or more data cells at data entry time, as shown in the following figure. The value for a row is the sum of the values in all of the expanded data cells in that row at the current column location. The current column changes as the current cell for which the rule is being evaluated changes within the grid.

For example, the value for row 1 is the sum of the values in the cells outlined in purple (1+5+9=15), when the rule is evaluated for any cell in column A with member Column_Member1. Similarly, the value for row 2 is the sum of values in cells outlined in blue (14+18=32) when the rule is evaluated for any cell in column A with member Column_Member2, and so on.



		A	
		Column_Member 1	Column_Member 2
1	Row_Member 1	1.0	2.0
	Row_Member 2	5.0	6.0
	Row_Member 3	9.0	10.0
2	Row_Member 4	13.0	14.0
	Row_Member 5	17.0	18.0

Cross Dim Member

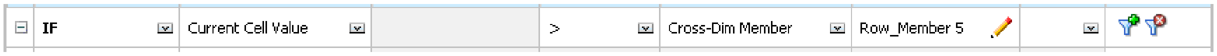
Action:

The action is performed 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.

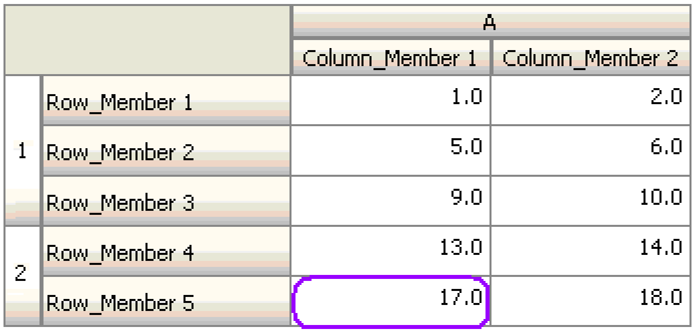
Condition Definition:



IF	Current Cell Value		>	Cross-Dim Member	Row_Member 5	

Condition Evaluation:

When the previous rule is applied at grid level, the rule is invoked on each cell in the data form, and the value in that cell is compared with the value in the cell outlined in purple. Thus, the cell at Row_Member 5->Column_Member 2 will turn red.



		A	
		Column_Member 1	Column_Member 2
1	Row_Member 1	1.0	2.0
	Row_Member 2	5.0	6.0
	Row_Member 3	9.0	10.0
2	Row_Member 4	13.0	14.0
	Row_Member 5	17.0	18.0

Member Name

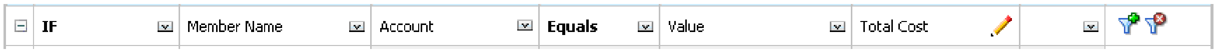
Action:

The action is performed 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 data form is created.

Operators:

The operators can be Equals, Not Equals, Contains, Starts With, or Ends With. The target value, which is the member name, is selected or entered free form.

Condition Definition:



IF	Member Name	Account	**Equals**	Value	Total Cost	

Member

Action:

The action is performed 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:

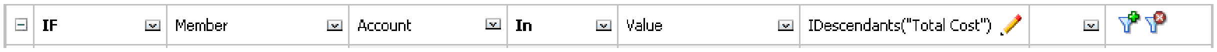
The available operators are In and Not In. The target value, which is the member, is selected or entered free form.

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.

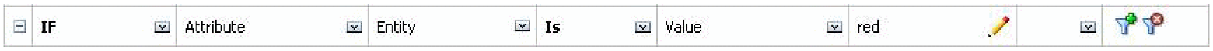
When using attribute values in data validation rules, keep in mind the way attributes are evaluated. If a rule references one or more attributes from one or more dimensions, they are evaluated as an OR for attribute values from the same attribute dimension, and as an AND for attributes from different 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 162.

Condition Definition for Member Source Type



IF	Member	Account	In	Value	IDescendants("Total Cost")	

Condition Definition for Attribute Source Type



IF	Attribute	Entity	Is	Value	red	

Account Type

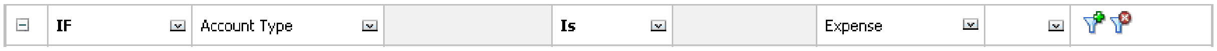
Action:

The action is performed 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:



IF	Account Type		Is		Expense	

Version Type

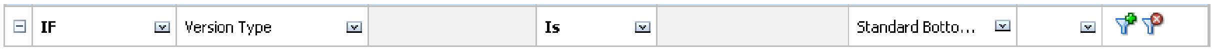
Action:

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:



IF	Version Type		Is		Standard Botto...	

Variance Reporting Type

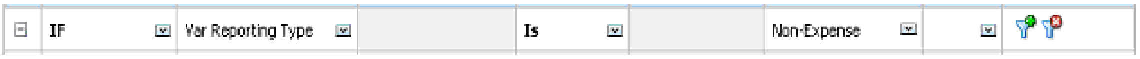
Action:

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:



IF	Var Reporting Type		Is		Non-Expense	

UDA

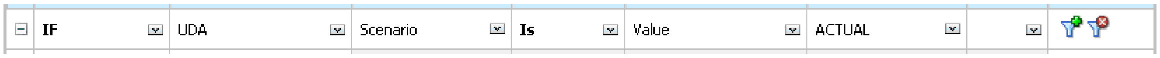
Action:

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:



IF	UDA	Scenario	Is	Value	ACTUAL	

Attribute

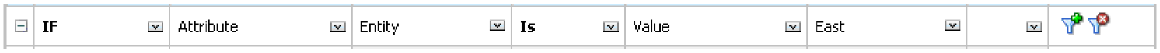
Action:

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:



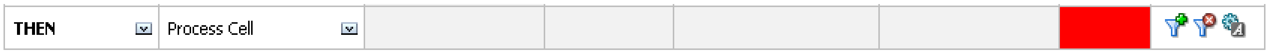
IF	Attribute	Entity	Is	Value	East	

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 172.

Condition Definition:



THEN	Process Cell					🟥	

For information on other conditions, see “Conditions Supported by the Rule Builder” on page 174.

Range Condition Values

The Check Range and Range conditions are used together. These conditions can be used in the data validation rule builder in a Then clause or standalone.

See:

- “Check Range” on page 182
- “Range” on page 182

For information on other conditions, see “Conditions Supported by the Rule Builder” on page 174.

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

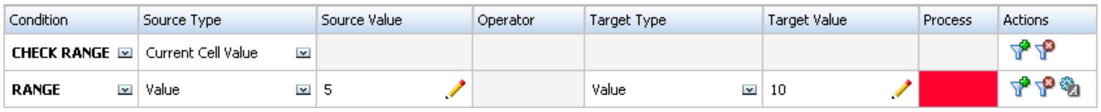
Action:

Defines a valid range for the value defined in the Check Range condition. This range includes all values that are >= the minimum value and < the maximum value. If the value specified in the Check Range condition is within this range, then the processing instructions defined by this condition are applied to the data cell on which the rule is being invoked. You can define multiple ranges of values and provide different processing instructions for each range.

Value:

The minimum and maximum values for the range can be defined using Cell Value, Current Cell Value, Row Value, Column Value, Cross-dimension Value, or by entering a free-form value. For example, the following rule ensures that the current cell value is >= 5 and < 10. If this condition is met, the cell is turned red.

Condition Definition:



Condition	Source Type	Source Value	Operator	Target Type	Target Value	Process	Actions
CHECK RANGE	Current Cell Value						
RANGE	Value	5		Value	10	🟥	

For information on other conditions, see "Conditions Supported by the Rule Builder" on page 174.

Data Validation Conditional Operators

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

- Numeric comparisons, using these operators: =, !=, <, <=, >, >=
- 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 58 Examples of Results for Starts With, Ends With, and Contains

Operator	Compare Value	Compare To Value
Starts With	2.0	2
	1234.0	12.0
	101.0	10
	2.0	2.0
	2.5	“2.”
	“YearTotal”	“Year”
Ends With	2.0	2.0
	2.0	2
	2.5	5
	2.5	“.5”
	“YearTotal”	“al”
	“YearTotal”	“Total”
Contains	2.0	2.0
	2.0	2
	2.5	5
	2.5	“.5”
	2.5	2.5
	23.567	3.5
	23.567	67
	23.567	“23.”
	23.567	“.56”
	“YearTotal”	“al”

Data Validation Rule Scenarios

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

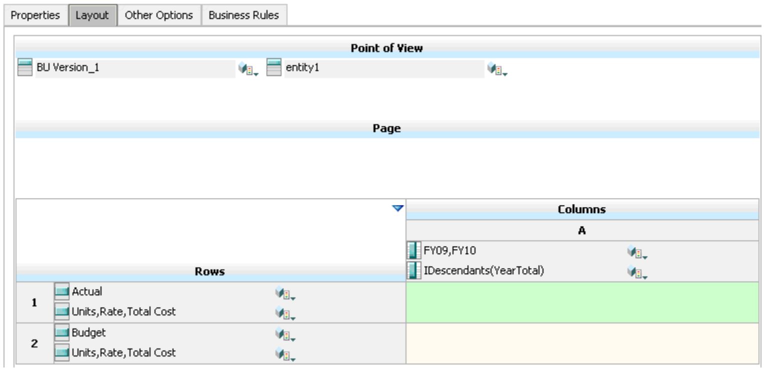
- “Scenario 1” on page 185
- “Scenario 2” on page 186

- “Scenario 3” on page 187
- “Scenario 4” on page 188
- “Scenario 5” on page 189

Scenario 1

John is hired by a company called Acme, Inc. as a consultant to design data 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 data form and adds a data validation rule at cell level using a cross-dimension member, as shown in the following figures.

Data Form Layout at Design Time:



Point of View

BU Version_1 | entity1

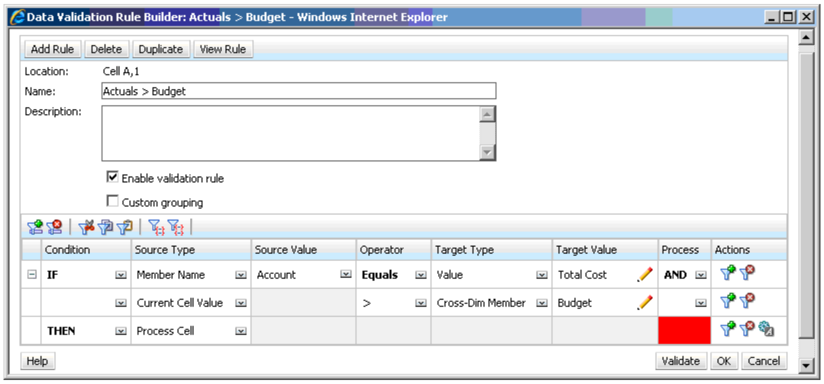
Page

Columns

		A
		FY09,FY10 IDescendants(YearTotal)
1	Actual Units,Rate,Total Cost	
2	Budget Units,Rate,Total Cost	

Rows

Data Validation Rule at Design Time:


Data Validation Rule Builder: Actuals > Budget - Windows Internet Explorer
Add Rule
Delete
Duplicate
View Rule

Location: Cell A,1

Name: Actuals > Budget

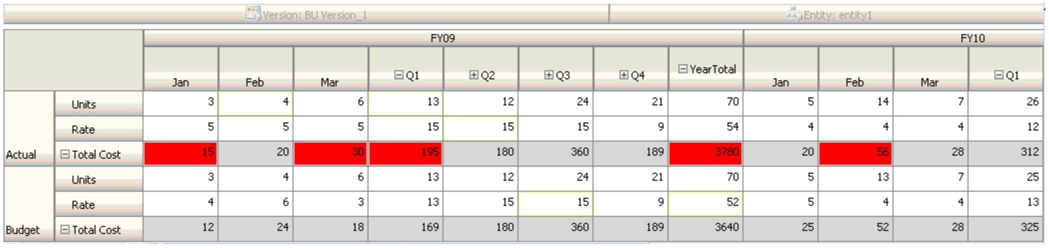
Description:

☑ Enable validation rule

☐ Custom grouping

Condition	Source Type	Source Value	Operator	Target Type	Target Value	Process	Actions
IF	Member Name	Account	Equals	Value	Total Cost	AND	
	Current Cell Value		>	Cross-Dim Member	Budget		
THEN	Process Cell					🟥	

Data Form at Data Entry Time with Data Validations Applied:



Version: BU Version_1 | Entity: entity1

		FY09								FY10			
		Jan	Feb	Mar	⊟ Q1	⊞ Q2	⊞ Q3	⊞ Q4	⊟ YearTotal	Jan	Feb	Mar	⊟ Q1
	Units	3	4	6	13	12	24	21	70	5	14	7	26
	Rate	5	5	5	15	15	15	9	54	4	4	4	12
Actual	⊟ Total Cost	15	20	30	195	180	360	189	3780	20	56	28	312
	Units	3	4	6	13	12	24	21	70	5	13	7	25
	Rate	4	6	3	13	15	15	9	52	5	4	4	13
Budget	⊟ Total Cost	12	24	18	169	180	360	189	3640	25	52	28	325

Tips:

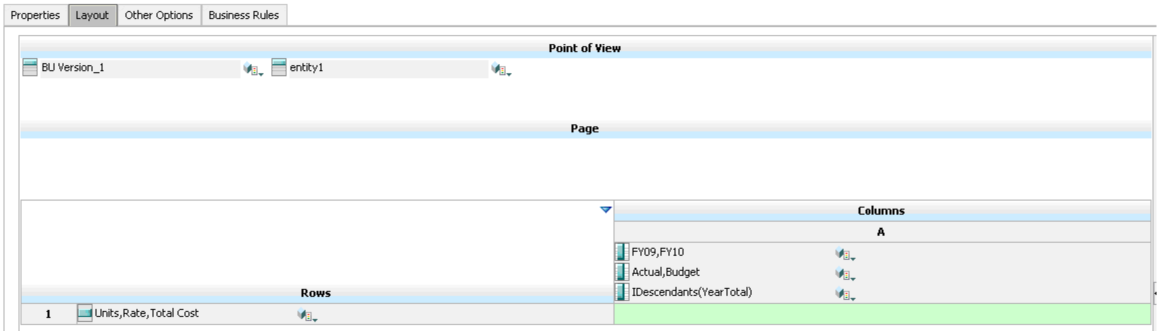
- John can split Total Cost into its own segment and apply the data validation rule at that segment for a slight performance gain. However, this would increase maintenance as new accounts and scenarios were added to the data form.
- If the requirements changed such that only the YearTotal Period in Actual 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 data form would be harder to maintain as new years were added.

For additional scenarios, see "Data Validation Rule Scenarios" on page 184.

Scenario 2

After reviewing the data 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 data form layout. However, he does not need to update the data validation rules. John updates the data form as shown in the following figure.

Data Form Layout at Design Time:



Point of View

BU Version_1 | entity1

Page

Columns

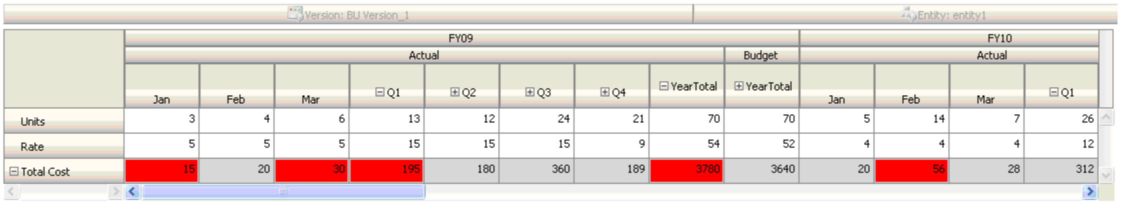
A

FY09,FY10
Actual,Budget
IDescendants(YearTotal)

Rows

1 Units,Rate,Total Cost

Data Form at Data Entry Time with Data Validations Applied:



Version: BU Version_1 | Entity: entity1

	FY09									FY10			
	Actual								Budget	Actual			
	Jan	Feb	Mar	⊟ Q1	⊞ Q2	⊞ Q3	⊞ Q4	⊟ YearTotal	⊞ YearTotal	Jan	Feb	Mar	⊟ Q1
Units	3	4	6	13	12	24	21	70	70	5	14	7	26
Rate	5	5	5	15	15	15	9	54	52	4	4	4	12
⊟ Total Cost	🟥 15	20	🟥 30	🟥 195	180	360	189	🟥 3780	3640	20	🟥 56	28	312

For additional scenarios, see “Data Validation Rule Scenarios” on page 184.

Scenario 3

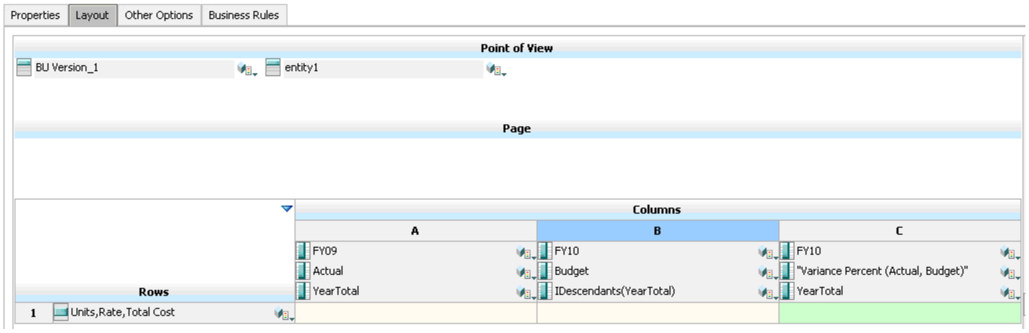
Following the successful rollout of these data forms, John is asked to implement the next policy, which is to ensure that this year’s Budget amounts are not significantly higher than previous year’s Actual amounts. If the difference is greater than 5%, then flag the difference in red.

John decides to use a member with a member formula to calculate the variance between this year’s Budget and the previous year’s Actual amount. He adds this member formula:

```
@varper(@Prior("Actual", 1, @Relative("Year", 0)), budget)/100;
```

John designs the data 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.

Data Form Layout at Design Time:



Point of View

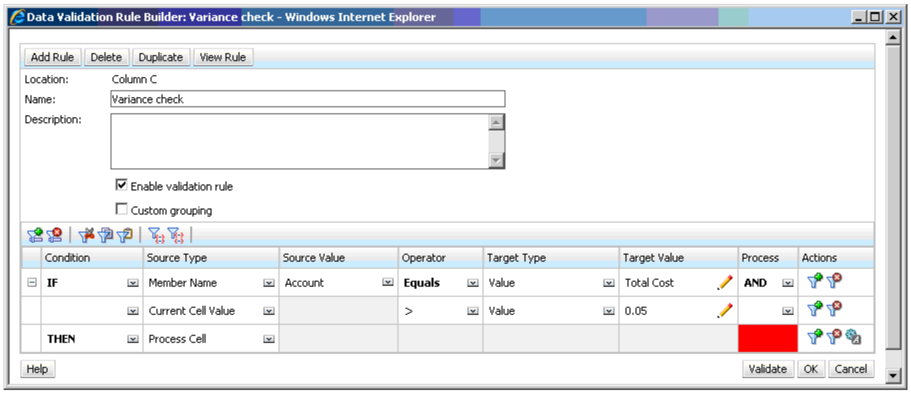
BU Version_1 | entity1

Page

Columns

Rows	A	B	C
	FY09	FY10	FY10
	Actual	Budget	"Variance Percent (Actual, Budget)"
	YearTotal	IDescendants(YearTotal)	YearTotal
1 Units,Rate,Total Cost			

Data Validation Rule at Design Time:



Data Validation Rule Builder: Variance check - Windows Internet Explorer
Add Rule
Delete
Duplicate
View Rule

Location: Column C

Name: Variance check

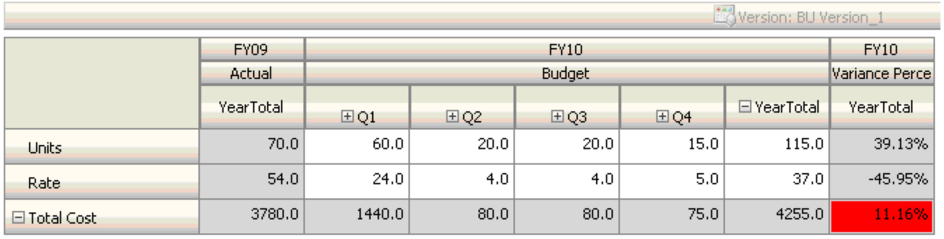
Description:

☑ Enable validation rule

☐ Custom grouping

	Condition	Source Type	Source Value	Operator	Target Type	Target Value	Process	Actions
⊟	IF	Member Name	Account	Equals	Value	Total Cost	AND	
		Current Cell Value		>	Value	0.05		
	THEN	Process Cell					🟥	

Data Form at Data Entry Time with Data Validations Applied:



Version: BU Version_1

	FY09	FY10					FY10
	Actual	Budget					Variance Perce
	YearTotal	⊞ Q1	⊞ Q2	⊞ Q3	⊞ Q4	⊟ YearTotal	YearTotal
Units	70.0	60.0	20.0	20.0	15.0	115.0	39.13%
Rate	54.0	24.0	4.0	4.0	5.0	37.0	-45.95%
⊟ Total Cost	3780.0	1440.0	80.0	80.0	75.0	4255.0	11.16%

Tips:

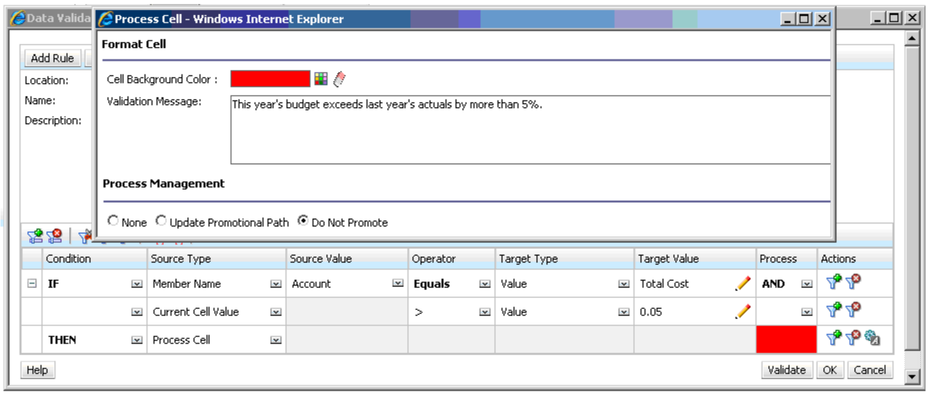
- If John is not allowed to change the outline, or if he experiences performance issues related to member formulas, he can use a formula column. See "Designing Data Forms with Formula Rows and Columns" on page 121.
- John defines the rule at the Variance Percent column for these reasons.
  - It improves 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 184.

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:



Format Cell

Cell Background Color : 🟥

Validation Message: This year's budget exceeds last year's actuals by more than 5%.

Process Management

○ None ○ Update Promotional Path ● Do Not Promote

	Condition	Source Type	Source Value	Operator	Target Type	Target Value	Process	Actions
⊟	IF	Member Name	Account	Equals	Value	Total Cost	AND	
		Current Cell Value		>	Value	0.05		
	THEN	Process Cell					🟥	

For additional scenarios, see “Data Validation Rule Scenarios” on page 184.

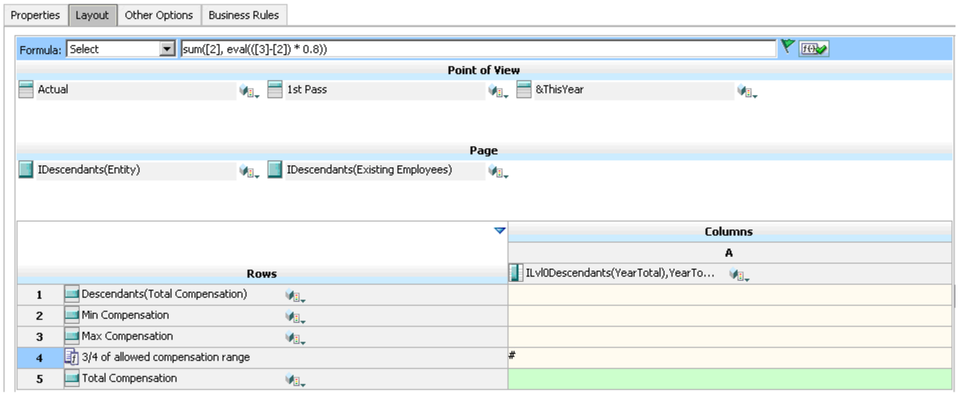
Scenario 5

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 data form in the Data Form Management dialog box. The data 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 data form, as shown in the following figures. Min Compensation and Max Compensation for the employee’s grade are calculated using member formulas.

Data Form Layout at Design Time:



Formula: Select sum([2], eval(([3]-[2]) * 0.8))

Point of View

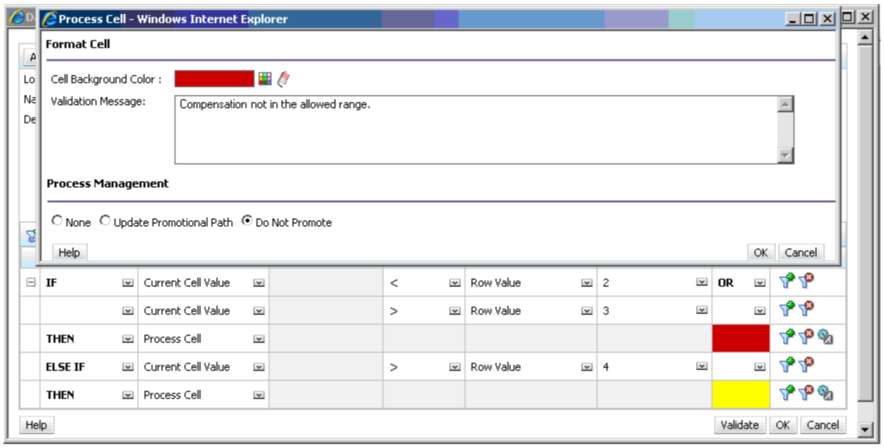
Actual | 1st Pass | &ThisYear

Page

IDescendants(Entity) | IDescendants(Existing Employees)

	Rows	Columns A ILvl0Descendants(YearTotal),YearTo...
1	Descendants(Total Compensation)	
2	Min Compensation	
3	Max Compensation	
4	3/4 of allowed compensation range	#
5	Total Compensation	

Data Validation Rule to Stop Promotion of Planning Units:



Process Cell - Windows Internet Explorer

Format Cell

Cell Background Color : 🟥

Validation Message: Compensation not in the allowed range.

Process Management

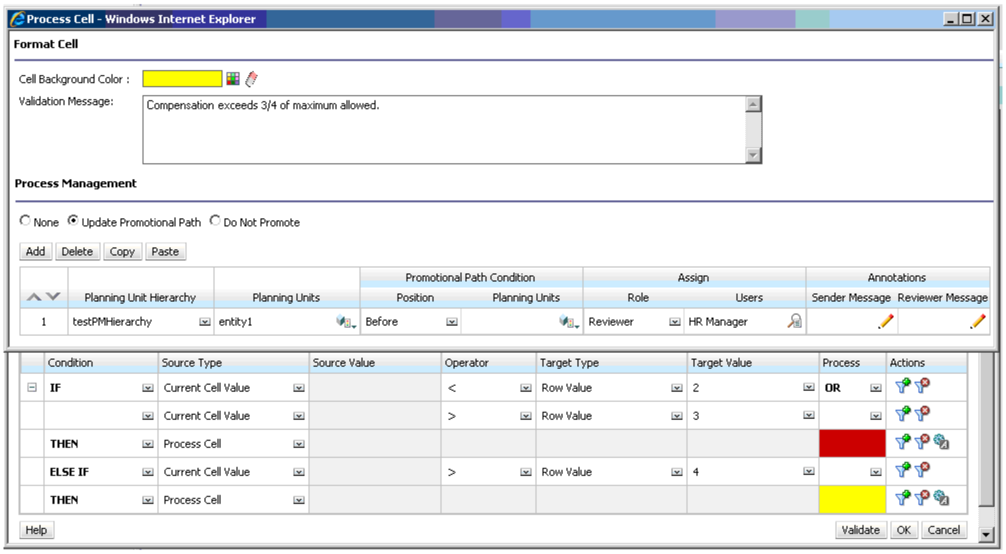
○ None ○ Update Promotional Path ● Do Not Promote

Help OK Cancel

IF	Current Cell Value		<	Row Value	2	OR	
	Current Cell Value		>	Row Value	3		
THEN	Process Cell					🟥	
ELSE IF	Current Cell Value		>	Row Value	4		
THEN	Process Cell					🟨	

Help Validate OK Cancel

Data Validation Rule to Add the Human Resources Manager as Reviewer:



Process Cell - Windows Internet Explorer

Format Cell

Cell Background Color : 🟨

Validation Message: Compensation exceeds 3/4 of maximum allowed.

Process Management

○ None ● Update Promotional Path ○ Do Not Promote

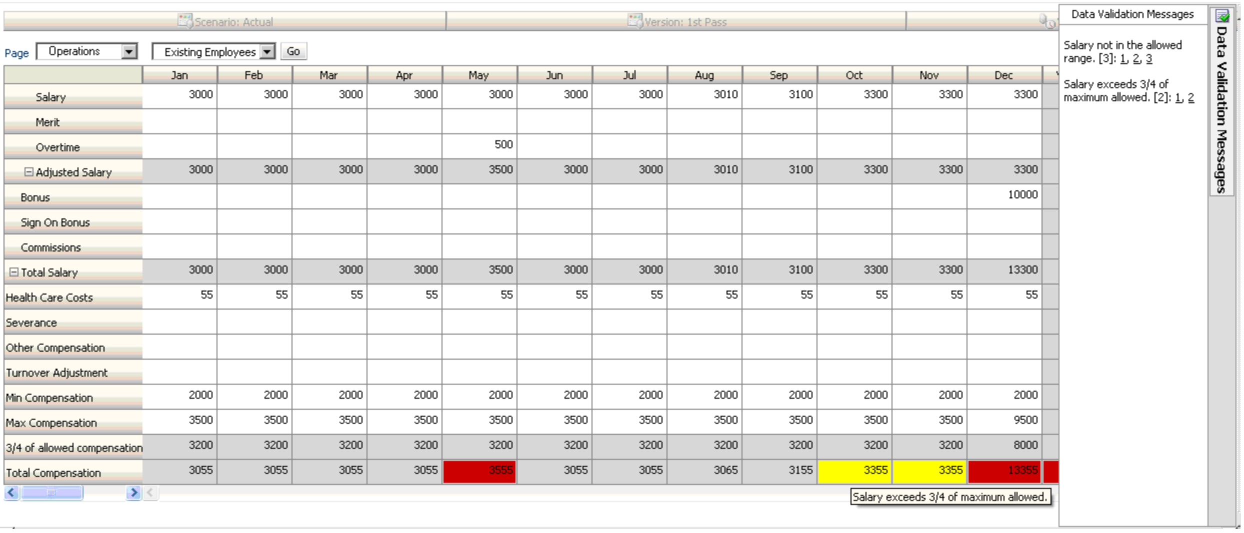
Add Delete Copy Paste

			Promotional Path Condition		Assign		Annotations	
	Planning Unit Hierarchy	Planning Units	Position	Planning Units	Role	Users	Sender Message	Reviewer Message
1	testPMHierarchy	entity1	Before		Reviewer	HR Manager		

Condition	Source Type	Source Value	Operator	Target Type	Target Value	Process	Actions
IF	Current Cell Value		<	Row Value	2	OR	
	Current Cell Value		>	Row Value	3		
THEN	Process Cell					🟥	
ELSE IF	Current Cell Value		>	Row Value	4		
THEN	Process Cell					🟨	

Help Validate OK Cancel

Data Form at Data Entry Time with Data Validations Applied and Validation Messages Shown:



Scenario: Actual

Version: 1st Pass

Page: Operations | Existing Employees | Go

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Salary	3000	3000	3000	3000	3000	3000	3000	3010	3100	3300	3300	3300
Merit												
Overtime					500							
⊟ Adjusted Salary	3000	3000	3000	3000	3500	3000	3000	3010	3100	3300	3300	3300
Bonus												10000
Sign On Bonus												
Commissions												
⊟ Total Salary	3000	3000	3000	3000	3500	3000	3000	3010	3100	3300	3300	13300
Health Care Costs	55	55	55	55	55	55	55	55	55	55	55	55
Severance												
Other Compensation												
Turnover Adjustment												
Min Compensation	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000
Max Compensation	3500	3500	3500	3500	3500	3500	3500	3500	3500	3500	3500	9500
3/4 of allowed compensation	3200	3200	3200	3200	3200	3200	3200	3200	3200	3200	3200	8000
Total Compensation	3055	3055	3055	3055	3555	3055	3055	3065	3155	3355	3355	13355

Salary exceeds 3/4 of maximum allowed.

Data Validation Messages

Salary not in the allowed range. [3]: 1, 2, 3

Salary exceeds 3/4 of maximum allowed. [2]: 1, 2

For additional scenarios, see "Data Validation Rule Scenarios" on page 184.

9 Managing the Budgeting Process

In This Chapter

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 199). 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 208.

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 data 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 E-mail for Approvals Notification

Application owners must specify an application's e-mail server before others can enable e-mail notification. After e-mail notification is configured, users receive e-mail when they become owners of planning units, or are specified as a user to notify. This feature is available for SMTP e-mail systems only; to specify the e-mail server, see "Specifying System Settings" on page 231.

If the e-mail server name is valid, the application owner becomes the source of all e-mail notifications. If the e-mail server name is not valid, nobody can enable e-mail notification.

Defining the Budgeting Process

Subtopics

- Approvals Roles
- Approvals Process

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 data 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 Hyperion Enterprise Performance Management System User and Role Security 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.

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 Administration, then Application, and then Reports.**

2 **Select Planning Unit Annotations.**

3 **Under Planning Units, select the Scenario, Version, and Entity combination for which you want to generate a report. If Custom is selected, click 🔍 to select the custom scenarios, versions, and entities.**

4 **Under Approvals Status, select status states.**

5 **Click Create Report.**

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 Hyperion 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

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 budget 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 budget process (see "Setting Planning Unit Hierarchy Name, Scope, and Template" on page 199).

4 Select the primary and subhierarchy members to include in the budget process (see "Selecting Planning Unit Hierarchy Members" on page 200).

5 Assign owners and reviewers for each stage of the budget process and create the planning unit promotional path (see "Assigning Planning Unit Owners and Reviewers" on page 202).

6 Click **Save.**

Setting Planning Unit Hierarchy Name, Scope, and Template

➤ To set up a planning unit hierarchy:

1 Select **Administration**, then **Approvals**, and then **Planning Unit Hierarchy**.

2 Take an action:

- To create a new hierarchy, click **Create**.
- To edit an existing hierarchy, select a planning unit hierarchy, and then click **Edit**.

3 Select **Approvals Dimension.**

4 In **Hierarchy Name**, provide the planning unit hierarchy name.

5 **Optional**: Provide a description of the planning unit hierarchy.

6 In **Enable Approvals**, select:

- All to add all planning units to the budget process.
- None to include no planning units in the budget process by default.

To add planning unit groups or individual planning units to the budget process, see "Selecting Planning Unit Hierarchy Members" on page 200.

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

  You can add individual planning units and planning units based on Parent member and generation criteria. See "Selecting Planning Unit Hierarchy Members" on page 200.

7 In **Approvals Template**, select:

- **Bottom Up** to use bottom-up budgeting (see "Bottom-Up Budgeting" on page 200).
- **Distribute** to use distributed budgeting (see "Distributed Budgeting" on page 200).
- **Free Form** to use free-form budgeting (see "Free-Form Budgeting" on page 200).

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 200).
- 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 199.

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.

Note: 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.

2 Use these buttons to control the planning unit hierarchy display:

- Click ⊞ to expand the display.
- Click ⊟ 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**, or click **Start** (first page), **Prev** (previous page), **Next**, or **End** (last page), and then **Go**.

3 **Optional:** For planning units not included in the default settings for the budget process, check the box to the left of the planning unit name to include the planning unit in the budget process.

4 **Optional:** Right-click a planning unit name to define subhierarchy members for the budget 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.

5 **Optional:** Add a secondary dimension to a planning unit included in the budget 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 budget process.

d. Check **Auto Include** if you want to add all members to the planning unit hierarchy that meet the selected criteria.

e. **Optional**: Click in **Selected Members** for the planning unit to refine the members it includes. Clear the check box next to any member you want to remove.

6 Take an action:

- Click **Next** or select **Assign Owners** to specify planning unit ownership (see "Assigning Planning Unit Owners and Reviewers" on page 202).
- Click **Save** and then **Finish** to save changes and close the planning unit hierarchy.

Assigning Planning Unit Owners and Reviewers

Planning unit ownership is inherited from the planning unit parents. Planning unit reviewers are also inherited. You can also explicitly specify planning unit owners and reviewers to assign owners and reviewers other than those planning units inherit.

➤ To assign planning unit reviewers and owners:

1 Take an action:

a. From **Primary and Subhierarchy Selection**, click **Next** or select **Assign Owners** to continue defining a planning unit hierarchy.

b. Select **Administration**, then **Approvals**, and then **Planning Unit Hierarchy**. Select a planning unit hierarchy, then click **Edit**, and then select **Assign Owners** to specify planning unit ownership.

2 Select a planning unit, then under **Owner**, click to select an owner.

Note: A planning unit can have only one owner.

3 Under **Reviewer**, click and select one or more users as planning unit reviewers.

Note: When using the Bottom Up or Distribute template, select 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.

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 **Action**, 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 199 to modify the planning unit hierarchy description or approvals default scope.
   - **Primary and Subhierarchy Selection**

     See "Selecting Planning Unit Hierarchy Members" on page 200 to select or remove members from approvals.
   - **Assign Owners**

     See "Assigning Planning Unit Owners and Reviewers" on page 202 to modify planning unit owners or reviewers
   - **Usage**

     See "Viewing Planning Unit Hierarchy Usage" on page 205 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 203.

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 205 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 data 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 data 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 **Data 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 data 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 169.
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 203.

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 205.

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 users 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 **Approvals**, and then **Planning Unit Hierarchy** to display the planning unit hierarchy list.

   Under the **Synchronized** column, planning unit hierarchies are labeled:

- **Yes**, if changes are synchronized with the planning unit hierarchy
- **No**, if changes are not synchronized with the planning unit hierarchy
- **Locked**, when 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.

2. Select a planning unit hierarchy listed as **No** 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. See "Working with the Outline Load Utility" on page 86 for information about the file format. After you create this file, you can copy its contents to an existing planning unit hierarchy (see "Importing Planning Unit Hierarchies" on page 207).

➤ 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.

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 207.

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.
4. For **Planning Unit Hierarchy with Ownership**, click **Browse** to select the exported planning unit hierarchy file to import.
5. Click **OK.**

   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 208). 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 169).

2 In the **Data Validation Rule Builder**, click to create promotional path conditions and specify the action taken when a budget calculation does not meet these conditions.

3 Select one action:

- **Update Promotional Path** to create a promotional path condition that adds reviewers or owners to the promotional path if the condition you set is triggered.
- **Do Not Promote** to prevent promoting a planning unit to the next reviewer. This action usually indicates invalid data in the budget.

4 In the **Process Cell** dialog box, beneath **Approvals**, click **Add** to add a promotional path condition.

Promotional path conditions are processed in the order they are listed. Use the ⌃ and ⌄ to move them up or down within the list.

5 **Optional**: To duplicate a promotional path condition:

a. Select a promotional path condition, and then click **Copy**.

b. Select the promotional path condition that will precede the copied one in the list, and then click **Paste** to add the copied promotional path condition.

6 **Optional**: Click **Delete** to remove a promotional path condition.

7 In **Planning Unit Hierarchy** and **Planning Units**, click to select the planning unit hierarchy and planning units the promotional path condition affects.

8 In **Promotional Path Condition**, specify the promotional path changes for the planning units selected in Promotional Path Condition. Select:

a. In **Position**, specify where the promotional path changes for the planning units selected in **Planning Units**. Select:

- **Before** to insert the alternate owner or reviewer before ownership reaches the planning units specified in **Promotional Path Condition**.
- **After** to insert the alternate owner or reviewer after ownership reaches the planning units specified in **Promotional Path Condition**.
- **Before and After** to insert the alternate owner or reviewer before and after ownership reaches the planning units specified in **Promotional Path Condition**.

Note: After you add promotional path conditions to a data validation rule, the planning unit promotional path displays the potential changes in planning unit ownership as an optional path above the defined promotional path. The display indicates where the path can change and how the alternate path rejoins the defined promotional path.

b. In the **Planning Units** column of **Promotional Path Condition**, click ▤ to select the planning units the action selected in **Position** affects.

> **Note:** The selected planning units must be ancestors of the planning units selected in the previous step. If you select planning units that are not ancestors, the promotional path condition is not evaluated when the planning unit hierarchy is validated.

9 **In the Assign column, specify the role, user type, and the user or UDA for each alternate owner, reviewer, and user to be notified if the rule exception occurs.**

a. In **Role**, select:

- **Owner** to select the alternate owner
- **Reviewer** to select the alternate reviewer
- **Notified Users** to select the user to be notified

b. In **Type**, select:

- **User Name** to specify the alternate user
- **From UDA** to select the dimension and to enter the prefix that was assigned to the UDA

c. In **Users**, specify:

- For **User Name**, the name of the alternate user or choose one from the user selector
- For **From UDA**, the dimension from the **Select a Dimension** drop down, and then specify the prefix that was used in the UDA to indicate that the UDA contains a user name (for example, ProdMgr:)

> **Note:** Ensure the prefix specified is the same as the prefix that was assigned while creating the UDA, including any spaces. For instance, if there is a space between the prefix and the username (ProdMgr Kim), ensure there is a space after ProdMgr.

10 **Optional: Provide messages for the users involved in the planning unit promotional path changes:**

a. Click ✎ in:

- **Sender Message** to enter the e-mail message text sent to the user promoting the planning unit when the promotional path changes.
- In **Reviewer Message** to enter the e-mail 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
- Editing Task Lists
- Linking Tasks to Planning Web 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. Users who are granted the Task List Access Manager role can assign task lists and tasks to other users.

➤ 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 data 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 211.
2. In the **Task List Folders** area, select the folder in which to create the task list folder.
3. Click **Create.**

4 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 211.

2 In the **Task List Folders** area, select the folder with the task list to rename.

3 Select the task list.

4 Click **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 211.

2 In the **Task List Folders** area, select the folder to rename.

3 Click **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 211.

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 211.

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 211.

2 In the **Task List Folders** area, select the folder in which to create the task list.

3 Above the **Task List** area, click **Create.**

4 Enter the task list name, and click **OK.**

5 To define the task list, see:

- “Adding Instructions to Task Lists” on page 213.
- “Adding and Defining Tasks” on page 214.

Adding Instructions to Task Lists

➤ To add instructions to task lists:

1 Open the **Manage Task Lists** page.

See “Managing Task Lists” on page 211.

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, and click **Edit.**

4 For **Edit Task List**, click **Instructions.**

5 Enter instructions for the task list.

6 Click **Save** and **Close.**

Adding and Defining Tasks

Subtopics

- Adding Tasks to Task Lists
- Setting Task Properties

After creating task lists, you can add and define tasks such as entering data in data forms and running required business rules.

Adding Tasks to Task Lists

You can set completion dates and alerts for tasks. Alerts display on task lists as colored circles:

- Green: On schedule
- Yellow: approaching due date
- Red: overdue

You can also set up e-mail 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 e-mail server (see “Specifying System Settings” on page 231).

➤ To add tasks to task lists:

1 Open the **Manage Task Lists** page.

See “Managing Task Lists” on page 211.

2 Select a task list, and click **Edit.**

3 Click **Add Child.**

4 For **Task**, enter a task name.

5 For **Type**, select the task type:

- URL Task: Opens a specified URL
- Web Data Form: Opens a data form
- Business Rule: Launches a business rule that you specify
- Manage Approvals: Starts the review process with a specified scenario and version
- Descriptive: Descriptive information

6 For **Instructions**, enter information that explains how to complete the task.

7 **Optional:** To enter a due date for the task, select **Due Date** and then select:

a. The month, day, and year. (You can change the date display format in Planning preferences. See “Specifying System Settings” on page 231.)

b. The hours, minutes, and AM or PM.

8 **Optional:** To send e-mail message for uncompleted tasks, select **Due Date**, and then:

a. Select **Repeat Every**, and enter a value.

b. Select the frequency for e-mail reminders.

9 **Optional**: To send e-mail messages after the alert date and before the due date, select **Alert**, and then:

a. Set the date and time to begin sending messages by selecting the month, day, year, hours, minutes, and AM or PM.

b. In the **Alert** area, select **Repeat Every** and enter a value.

c. Select the frequency for e-mail reminders.

10 **Optional**: To make task completion dependent on completing a primary task, select **Dependency.**

11 Click **Save** and **Close.**

12 Set properties (see "Setting Task Properties" on page 215).

Setting Task Properties

You must set properties for all task types except Descriptive tasks.

➤ To set task properties:

1 Open the **Manage Task Lists** page.

See "Managing Task Lists" on page 211.

2 Select a task list, then click **Edit.**

3 Select a task, then click **Edit.**

4 Click **Property.**

5 In **Property**:

Table 59 Setting Task Properties

Type of Task	Action
URL	Enter a fully qualified URL to associate with this task, such as `http://www.company_name.com`. Optionally, select Use Single Sign On to enable users to open a URL for another product that accepts single sign-on (see the *Oracle Hyperion Enterprise Performance Management System Security Administration Guide*). To link to Financial Reporting in the EPM Workspace, single sign-on is not required. Instead, include the ObjectID to link to (see the *Oracle Enterprise Performance Management Workspace Developer's Guide*).
Data Form	For Data Form Folder, select the folder containing the data form associated with this task. For Data Form, select the data form for users to complete. **Optional**: Select **Set Page Member Defaults** to select the member from each dimension to display as the default when the task is first opened. After you select this option, you can select the members for the page dimensions. The page member defaults apply until a user updates the data 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.
Business Rule	For Plan Type, select the plan type associated with the business rule to execute. For Business Rules, select the business rule to execute.
Approvals	For Scenario and Version, select the scenario and version users will work in.

For Descriptive tasks, no updates are required.

6 Click **Save**.

7 Click **Close**.

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 e-mail reminder messages.

➤ To edit tasks:

1 Open the **Manage Task Lists** page.

See "Managing Task Lists" on page 211.

2 Select a task list, and click **Edit**.

3 Select a task, and click **Edit**.

4 In **Edit Task**:

- Modify the task name.
- Select another task type:
  - URL Task: Opens a specified URL
  - Web Data Form: Opens a data form
  - Business Rule: Launches a specified business rule
  - Manage Approvals: Starts the review process with a specified scenario and version
  - Descriptive: Describes an action that users must take

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 e-mail 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 e-mail reminders.

8 **Optional**: To send e-mail 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 e-mail 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.** See "Setting Task Properties" on page 215.

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 211.

2 Select a task list, and click **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 211.

2 Select a task list, and click **Edit.**

3 Select a task, then click the up or down arrow.

4 Click **Save** and **Close.**

➤ To cut and paste tasks:

1 Open the **Manage Task Lists** page.

See "Managing Task Lists" on page 211.

2 Select a task list, and click **Edit.**

3 Select a task, then click **Cut.**

4 To move the task to a new position, select the task to appear above it.

5 Click **Paste.**

6 Click **OK.**

Moving Task Lists

➤ To move task lists:

1 Open the **Manage Task Lists** page.

See “Managing Task Lists” on page 211.

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 211.

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 211.

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 **OK**.

Deleting Tasks

➤ To delete tasks:

1 Open the **Manage Task Lists** page.

See "Managing Task Lists" on page 211.

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

3 Select the tasks to delete, and click **Delete**.

4 Click **OK**.

Deleting Task Lists

➤ To delete task lists:

1 Open the **Manage Task Lists** page.

See "Managing Task Lists" on page 211.

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

3 Click **Delete**.

4 Click **OK**.

Linking Tasks to Planning Web Pages

Use Copy Document Link to link tasks in task lists to application pages. You can copy and paste the URL address from a page in the Planning application to a task. The task list user can access the Planning page within the task list.

➤ To copy URLs to task lists:

1 Access the page in the Planning application to which to link a task.

2 Select **Tools**, then **Copy Link**.

3 Open the task list to which to link a task.

4 Select the task to which to link the copied Planning page and select **Edit**.

5 Click **Property**.

6 Use the browser to paste the information. For example, in the Microsoft Internet Explorer menu, select **Edit**, then **Paste**.

7 **Click Save.**

8 **Click Close.**

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. Interactive users and planners can access task lists in Basic mode based on access permissions. Users granted the Task List Access Manager role can assign task lists and tasks to other users.

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 211.

2 **Select the task list to modify.**

3 **In the Task List area, click Assign Access.**

- **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**.
- **Optional:** To remove deprovisioned or deleted users or groups from the Planning database to conserve space, click **Remove Nonprovisioned Users/Groups**.

4 **Click Add Access.**

5 **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.

6 **For Type of Access, select how users or groups can use the task list:**

- **Assign**: View and use
- **Manage**: Modify
- **Manage and Assign**: View, use, and modify
- **None**: No access

7 **Click Add.**

8 Click **Close.**

Changing Access to Task Lists

➤ To change access to task lists:

1 Open the **Manage Task Lists** page.

See “Managing Task Lists” on page 211.

2 Select the task list to modify, then click **Assign Access.**

- 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**.
- Optional: To remove deprovisioned or deleted users or groups from the Planning database to conserve space, click **Remove Nonprovisioned Users/Groups**.

3 Select users or groups, and click **Edit Access.**

4 For **Type of Access**, select:

- **Assign**: View and use
- **Manage**: Modify
- **Manage and Assign**: View, use, and modify
- **None**: No access

5 Click **Set.**

6 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 211.

2 Select the folder and task list to modify, then click **Assign Access.**

- Optional: To migrate a user or group's changed identity or their position in the user directory from Oracle's Hyperion® Shared Services Console to Planning, click **Migrate Identities**.
- Optional: To remove deprovisioned or deleted users or groups from the Planning database to conserve space, click **Remove Nonprovisioned Users/Groups**.

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

4 Click **OK.**

5 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 45.

➤ To launch the `TaskListDefUtil` utility:

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

```
TaskListDefUtil [-f:passwordFile] import|exportFILE_NAME|
TASK_LIST_NAME|-all SERVER_NAME USER_NAME APPLICATION
```

Parameter	Purpose	Required?
`[-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 44.	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:

```
TaskListDefUtil.cmd export -all localhost admin APP1
```

- To import all task list definitions:

```
TaskListDefUtil.cmd import -all localhost admin APP1
```

Copying Data

You can copy plans from one dimensional intersection to another, including relational data and supporting detail. For example, you can copy *Budget, FY10, Final* to *Forecast, FY11, First Draft*. Notes:

- Selected Copy Data settings are preserved for the current session only.
- Copied dimension members must be present in the selected plan types.
- Data must be copied into cells that can accept data. For example, you cannot copy data into read-only or dynamic cells.
- You can copy account annotations, supporting detail, and cell text. You cannot copy planning unit annotations.
- You cannot use this feature with attributes, so do not select attributes to be copied.
- 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**

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 Online Help*.

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 the *Oracle Hyperion Planning User's Online Help*.

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 Hyperion 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 241.

➤ 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 🔍. 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 Online Help*.

9 To clear cell details for another plan type, select another plan type in step 2, and repeat the procedure.

10 Working With Applications

In This Chapter

Managing Data Forms and Folders

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

Task	Topic
Create folders	See "Creating Folders" on page 227
Move folders	See "Moving Folders" on page 228
Delete folders	See "Deleting Folders" on page 228
Create data forms	See "Creating Simple Data Forms" on page 124
Assign access to data forms and folders	See "Assigning Access to Data Forms and Folders" on page 54
Move data forms	See "Moving Data Forms and Ad Hoc Grids" on page 151
Delete data forms	See "Deleting Data Forms" on page 151
Rename folders	See "Renaming Folders" on page 229

To view all data forms or business rules in a Calculation Manager folder, click the folder's name in the lefthand folders area. To select all the data forms, select the check box at the top of the forms list.

Creating Folders

Use folders to hierarchically organize data 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 (if the application uses Calculation Manager as the calculation module) **CalcMgrRules**

➤ To create folders:

1 Perform an action:

  - For data form folders: Select **Administration**, then **Manage**, and then **Data 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, data forms, and Calculation Manager business rules within them are also moved.

➤ To move folders:

1 For data form folders: Select **Administration**, then **Manage**, then **Data 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 data form folders: Select **Administration**, then **Manage**, then **Data 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 delete folders:

1 For data form folders: Select **Administration**, then **Manage**, and then **Data Forms and Ad Hoc Grids.**

For 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 the *Oracle Hyperion Planning User's Online Help*.

See:

- "Setting Personal Preferences" on page 230
- "Setting Application Defaults" on page 230
- "Specifying System Settings" on page 231
- "Limiting Use of Applications" on page 232
- "Specifying Custom Tools" on page 233
- "Setting Display Options" on page 233
- "Setting Printing Options" on page 234

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

Show Option	Description
Current Application Defaults	Accesses the Application Settings tab and Display Options tab. The values set on these tabs become application defaults. Users can override defaults, and can revert to defaults by selecting Use Application Defaults on preference tabs. See the *Oracle Hyperion Planning User's Online Help*.
Advanced Settings	Accesses the System Settings tab and Custom Tools tab. See "Specifying System Settings" on page 231 and "Specifying Custom Tools" on page 233.

Setting Personal Preferences

All users can set personal preferences in the current application. For Application settings, Display Options, and Printing Options, see the *Oracle Hyperion Planning User's Online Help*.

➤ To set personal preferences:

1 Select **File**, then **Preferences.**

2 Click **Planning**, then perform an action:

- Select **Application Settings** to set e-mail 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 data 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 data form by setting a user variable. See "Managing User Variables" on page 149.

Selecting **Use Application Default** resets the value to the current application default.

Setting Application Defaults

Administrators can specify defaults for the current application. Users can override application defaults by setting preferences. They can restore application defaults by selecting Use Application Default where available. For Application Settings and Display Options, see the *Oracle Hyperion Planning User's Online Help*.

➤ To set application defaults:

1 Select **Administration**, then **Application**, then **Settings.**

2 For **Show**, select **Current Application Defaults.**

3 Click **Go.**

4 Perform one action:

- Select **Application Settings** to set e-mail options for task lists, approvals, the application owner, and the job console, and set options for planning units and attribute dimensions. You can also 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.
- Select **Display Options** to set options for number formatting, page selection, search, indentation of members, consolidation operators, warnings for large data forms, the number of dimensions to show on a page, UI theme, text size, and date format. Another option sends you directly to the page you most recently visited, the next time you log on.

5 Click **Save.**

These settings become application defaults, which users can select with Use Application Defaults.

Specifying System Settings

Only administrators can specify application-wide settings. Only the application owner can set the E-mail Server and Password for Synchronizing With Essbase.

➤ To specify system settings:

1. Select **Administration**, then **Application**, then **Settings.**
2. For **Show**, select **Advanced Settings.**
3. Click **Go.**
4. Select **System Settings.**
5. Set options:

Table 60 System Settings

Option	Description
E-mail Server	The server name that hosts e-mail services for application users, such as `mail.oracle.com`
E-mail Character Set	The character set for e-mail messages: **UTF-8**: Unicode encoding format **Regional Setting**: The system's regional setting
Shared Services URL	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 232).
Display Users' Full Names	**Yes**: Show full names (such as Victoria Hennings) and user names (such as VHennings). **No**: Do not display. With this selected, two users cannot have identical full names.
Enable Use of the Application for	Determine whether users can access the application in maintenance mode, such as during backups. See "Limiting Use of Applications" on page 232.
Enable Display of Substitution Variables	Set how substitution variables display in the Member Selection dialog box when users respond to runtime prompts in business rules: ● **Display All**: Display all substitution variables ● **Display None**: Do not display substitution variables ● **Enable Filtering**: Display only substitution variables that are valid for the runtime prompt
Select User	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.)

Option	Description
Calculation Module	Select whether to use **Business Rules** or **Calculation Manager** as the calculation module. ● If, after creating or modifying business rules, you switch from **Business Rules** to **Calculation Manager**, you can migrate the business rules through the Calculation Manager user interface. ● If, after creating or modifying business rules, you switch from **Calculation Manager** to **Business Rules**, the rules are not accessible in Business Rules. You can, however, switch back to the **Calculation Manager** calculation module to use the rules. **Tip:** To evaluate which calculation module you prefer before committing to it for multiple applications, first experiment with one calculation module on one application. **Note:** Performance Management Architect applications must use Calculation Manager, so for those applications, the **Calculation Manager** option is selected and cannot be changed.

6 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 Applications

Administrators can grant and withdraw access to applications. If users are logged on to the application and administrators withdraw their access, users receive a message and are forced off the system. Restricting use of an application as described here has no effect on Smart View users.

➤ To limit use of applications:

1 Select **Administration**, then **Application**, then **Settings**.

2 For **Show**, select **Advanced Settings**.

3 Click **Go**.

4 Select **System Settings**.

5 For **Application Maintenance Mode**, select an option for **Enable Use of the Application For**:

- **All users**: All users can log on or continue working with the application.
- **Administrator**: Only other 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.

6 If your selection is more restrictive than the current setting, click **OK.**

7 Click **Save.**

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 Data Form Precision, Display Properties, and Other Options" on page 129.

Mapping Applications for Reporting

Subtopics

- About Reporting Applications
- Creating Reporting Applications
- Mapping Planning Applications to Reporting Applications
- Defining Application Mappings
- Defining Dimension Mappings
- Setting the Point of View
- Validation Rules for Default Members
- Pushing Data to Reporting Applications
- 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 Reporting Applications

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 source Planning applications and reporting applications 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 Reporting Applications

➤ To create a reporting application:

1 **If it does not exist, create the target reporting application.**

Reporting applications can be either Essbase block storage or aggregate storage databases. 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 applications, 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 Planning Applications to Reporting Applications" on page 235.
- Push data from the source to the target application.

  See "Pushing Data to Reporting Applications" on page 238.
- Report on the data in the reporting application.

Mapping Planning Applications to Reporting Applications

➤ To map a Planning application to a reporting application:

1 **Create the reporting application.**

See "Creating Reporting Applications" on page 235.

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 236.
- 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 Reporting Applications" on page 238.
- To update the mapping for Smart Lists, see "Synchronizing Smart Lists in Reporting Applications" on page 310.

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 240, "Editing an Essbase Server for Reporting Applications" on page 240, or "Removing an Essbase Server for Reporting Applications" on page 241.

5 Click **Next.**

See "Defining Dimension Mappings" on page 236.

Defining Dimension Mappings

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 237).

On the **Map Dimensions** tab, you map the Planning dimensions on the left to the reporting application members on the right.

➤ 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 310.

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 237.
- **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 238.

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 🔍, 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 238.

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 Reporting Applications

After setting up application mappings, you can push data to reporting applications. 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's status. See “Mapping Planning Applications to Reporting Applications” on page 235.

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 45.

➤ To launch the `PushData` utility:

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

```
PushData [-f:passwordFile] /U:username /A:sourceApplication /
M:applicationMapping [/C]
```

Parameter	Description
`[-f:passwordFile]`	**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 44.
`/U:username`	The name of the Planning administrator
`/A:sourceApplication`	The name of the source Planning application from which to push data
`/M:applicationMapping`	The name of the application mapping that defines the push data range (see "Mapping Applications for Reporting" on page 234)
`[/C]`	**Optional**: Clears the data range in the target reporting application before pushing data

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 45.

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 applications to reporting applications 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 applications that contain 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 by Essbase 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.**
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 255).

➤ 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.

2 In **Server Name**, type the server name.

3 In **User Name**, type your user name.

4 In **Password**, type your password.

5 **Optional**: To test the connection to the Essbase server, click **Validate Connection.**

6 Click **OK.**

Removing an Essbase Server for Reporting Applications

➤ To remove an 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 241.
- Supporting detail associated with scenarios. See "Deleting Supporting Detail Associated With a Scenario" on page 242.

  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 224.

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 224.

➤ To delete account annotations associated with account names:

1 **Stop the Web application server.**

2 **Update the SQL queries section of the** `aadelete.sql` **file corresponding to your type of relational database by substituting the name of the account whose annotations to delete.**

3 **Run queries in the** `aadelete.sql` **file appropriate for your relational database.**

Example: deleting account annotations for `Account1`:

```
DELETE

FROM HSP_ACCOUNT_DESC

WHERE ACCOUNT_ID=(SELECT OBJECT_ID FROM HSP_OBJECT

WHERE OBJECT_NAME='ACCOUNT1')

INSERT INTO HSP_ACTION

(FROM_ID, TO_ID, ACTION_ID, OBJECT_TYPE, MESSAGE, ACTION_TIME,
PRIMARY_KEY) VALUES (0,0,2,18,NULL,GETDATE(),NULL)
```

Deleting Supporting Detail Associated With a Scenario

You can use the `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 224.

➤ 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:

```
DELETE

FROM HSP_COLUMN_DETAIL_ITEM

WHERE DETAIL_ID IN

  (SELECT DETAIL_ID

   FROM HSP_COLUMN_DETAIL

   WHERE DIM1 =

    (SELECT OBJECT_ID
     FROM HSP_OBJECT
     WHERE OBJECT_NAME ='ACTUAL'));
DELETE
FROM HSP_COLUMN_DETAIL
WHERE DIM1 =
  (SELECT OBJECT_ID
   FROM HSP_OBJECT
   WHERE object_name ='Actual');
```

11

Working with Menus

In This Chapter

Creating and Updating Menus

Administrators can create right-click menus and associate them with data forms, enabling users to click rows or columns in data forms and select menu items to:

- Launch another application, URL, or business rule, with or without runtime prompts
- Move to another data form
- Move to Manage Approvals with a predefined scenario and 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 data forms, use Other Options to select menus available for Data Form menu item types. As you update applications, 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: URL, Data Form, Business Rule, Manage Approvals, or Menu Header.

➤ 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 **Add Sibling**.
- To edit menu items, click **Edit**.
- To delete menu items, click **Delete**.
- To change the order of menu items within the same level, click the Up or Down Arrow. You can move multiple items.

  Use **Edit Menu Item** to define the menu item properties.

5 Click **Save.**

Click **Save As** to save the current selections under a new menu name.

Adding or Changing Menu Items

➤ To define menu items:

1 Select **Administration**, then **Manage**, then **Menus.**

2 Select a menu and click **Edit.**

3 Select the menu item and click **Edit** or **Add Sibling.**

4 Define the menu item:

Table 61 Menu Items

Item	**Description**
Menu Item	Enter a unique name containing only alphanumeric and underscore characters, with no special characters or spaces
Label	Enter text to 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.
Icon	**Optional**: In context of the Planning server, enter the path and filename to a graphic to display by the menu. For example: `/HyperionPlanning/Images/globe.gif`

Item	Description
Type	Select the menu item type to determine available Properties. No properties are available for Menu Header. ● **Data Form**: Launch a selected data form. The member selection context for the member, page, and POV is retained when users right-click in the source data form. If the target data 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 data 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 data 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 data form.

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

Type	Options
Data Form	a. From **Data Form Folder**, select the folder containing the destination data form. b. From **Data Form**, select the data form.
URL	a. For **URL**, enter the complete URL to which to direct the user. For example: `http://`*`server name`*`/HFM/Logon/HsvLogon.asp`. b. Select **Use Single Sign-on** to append the SSO token to the URL. c. Select **Include Context in URL** to include the context.
Business Rule	a. For **Plan Type**, select the plan type for which the business rule is available. b. For **Business Rules**, select the business rule to launch. c. From **View Type**, select how to display runtime prompt pages: ● **Classic View**: Use the default Planning view ● **Streamline View**: Display each runtime prompt on a different line d. **Optional**: For **Window Title**, enter a title to display instead of Runtime Prompts. e. **Optional**: For **OK Button Label**, enter the text to display for the OK button. f. **Optional**: For **Cancel Button Label**, enter the text to display for the Cancel button. g. **Optional**: Select **Launch in a Separate Window** to launch the business rule in a separate browser window.
Manage Approvals	Specify the planning unit to which the user is directed by selecting a scenario and a version.
Previous Form	Enter the name of the menu item that will return the user to the previous data form.

6 Click **Save.**

12 Working with Classic Application Administration

In This Chapter

About Creating Applications with Performance Management Architect and Classic Administration

You can create applications using Performance Management Architect or Classic 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 Classic administration, you can select Administration, then Manage, then Smart Lists to use the feature within Planning. Classic application administration tasks are described in this chapter. To use Classic application administration, you must be assigned the appropriate roles, as described in the *Oracle Hyperion Enterprise Performance Management System User and Role Security Guide*.

For information on setting up applications for Oracle Hyperion Public Sector Planning and Budgeting, Fusion Edition, see the *Oracle Hyperion Public Sector Planning and Budgeting User's Guide*.

Setting Up Classic Applications

For Classic Planning applications, use the Classic Application Wizard to create and delete applications, and register with Shared Services. When you create Classic applications, you can set them up to calculate business rules using Business Rules or Calculation Manager. After you set up the calculation module, you can change it as described in “Specifying System Settings” on page 231.

➤ To set up Classic applications:

1 **Perform one task:**

- From Planning, select **Administration**, then **Application**, then **Create**, **Delete**, or **Register**. Or, select **Administration**, then **Data Source**.
- From EPM Workspace, select **Navigate**, then **Administer**, then **Classic Application Administration**, and then **Planning Administration**. Then select **Create Application**, **Manage Data Source**, **Delete Application**, or **Register Application**.

2 **See these topics to complete the task:**

- "Creating Applications" on page 250.
- "Managing Data Sources" on page 250.
- "Deleting Applications" on page 261.
- "Registering Applications" on page 260.

Creating Applications

➤ To create and update Classic applications:

1 **Start the Classic Application Wizard (see "Setting Up Classic Applications" on page 249).**

2 **Define the application by completing information in the tabs. See:**

- "Selecting Application Information" on page 256
- "Managing Data Sources" on page 250
- "Setting up Currencies" on page 259
- "Specifying Plan Types" on page 259
- "Reviewing Application Information" on page 260

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 Classic Application Wizard to create, update, test, and delete data sources for Classic Planning applications, see:

- "Creating Data Sources" on page 251
- "Editing Data Sources" on page 252
- "Checking Connections" on page 253
- "Deleting Data Sources" on page 253

For Planning applications created in Performance Management Architect, you manage data sources in Performance Management Architect. See the *Oracle Hyperion Enterprise Performance Management Architect Administrator's Guide*.

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

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 Classic Planning applications, 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 Classic Planning applications:

1. **In the Classic Application Wizard, click Manage Data Source, and then click Create Data Source.**

   See "Setting Up Classic Applications" on page 249.

   Within Planning, you can also select **Administration**, then **Data Source**, then **Create**.

2. **Enter the data source name and description.**
3. **Specify application database details:**
   - **Select Database Platform**: Select a relational database
   - **Server**: Server hosting the database
   - **Port**: Port (for default ports, see *Oracle Hyperion Enterprise Performance Management System Installation Start Here*)
   - **Database**: Database name
   - **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 Hyperion 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.

5 Click **Validate Database Connection**, and fix any issues that are noted.

6 Specify Essbase server details:

- **Server**: Server name
- **User**: Server username
- **Password**: Server password

Note: If the default port number for Essbase, 1423, was changed to another number during Essbase configuration, the Essbase server must have the format `serverName:newPortNumber`. For example, if the Essbase server port number is set to 1400 during configuration and the Essbase server is run locally, the server name is `localhost:1400`.

7 Click **Validate Essbase Connection**, and fix any issues that are noted.

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 **Finish.**

10 At the message that the data source was created successfully, click the X to close the message.

11 Use the EPM System Configurator to select a cluster (instance) to use for the application. See the *Oracle Hyperion Enterprise Performance Management Architect Administrator's Guide* or Online Help.

Editing Data Sources

For Classic Planning applications, 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 Classic Application Wizard, click **Manage Data Source.**

See "Setting Up Classic Applications" on page 249.

2 Select a data source, and click **Edit Data Source.**

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 Hyperion 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 Database Connection**, and fix any issues that are noted.

7 Specify Essbase server details:

- **Server**: Server name
- **User**: Server username
- **Password**: Server password

8 Click **Validate Essbase Connection**, and fix any issues that are noted.

9 **Optional**: To set the application to Unicode mode, select **Unicode Mode**.

For more information, see the *Oracle Essbase Database Administrator's Guide*.

10 Click **Finish**.

11 At the message that the data source was updated successfully, click the X to close the message.

12 Use the EPM System Configurator to select an instance (cluster) to use for the application. See the *Oracle Hyperion Enterprise Performance Management Architect Administrator's Guide* or Online Help.

Checking Connections

You can test the connections to the database or Essbase.

➤ To check connections:

1 In the Classic Application Wizard, click **Manage Data Source**.

See "Setting Up Classic Applications" on page 249.

2 Click a button for the connection to test:

- **Check Database Connection**.
- **Check Essbase Connection**.

3 At the message that the connection is successful, click the X to close the message.

Deleting Data Sources

For Classic Planning applications, you can use the Classic Application Wizard to delete data sources that are not associated with an application.

➤ To delete data sources:

1 In the Classic Application Wizard, click **Manage Data Source.**

  See "Setting Up Classic Applications" on page 249.

2 Select the data source to delete, and click **Delete Data Source.**

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 Classic Planning.

- For updating references to rehosted relational databases and Essbase servers, see "Updating References to Data Sources" on page 254.
- For updating references to rehosted Essbase servers that have reporting applications, see "Updating References to Reporting Essbase Servers" on page 255.
- For upgrading Planning applications created in Performance Management Architect or Classic Planning, see "Upgrading Applications" on page 256.

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 Classic Application Wizard, click **Upgrade Wizard.**

2 On the **Update Data Sources** page, 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.
  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:

- 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** page (see "Upgrading Applications" on page 256), or click **Cancel** to return to the Classic Application Wizard launch page.

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

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 Classic Application Wizard, click **Upgrade Wizard**.

2 On the **Update Reporting Essbase Servers** page, 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:

- 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 **Previous** to switch to the **Upgrade Applications** page (see "Upgrading Applications" on page 256), or click **Cancel** to return to the Classic Application Wizard launch page.

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 in Performance Management Architect or Classic Planning. Instructions for this procedure are available in the *Oracle Hyperion EPM 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 254 and "Updating References to Reporting Essbase Servers" on page 255.

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*.

Selecting Application Information

Specify the application name and description, register the application with Shared Services, select the data source, and select Business Rules or Calculation Manager to calculate business rules. For information about Hyperion Calculation Manager, see the *Oracle Hyperion Calculation Manager Designer's Guide*.

A default instance (cluster) is set up when you install and configure Planning. To update clusters with the Oracle's Hyperion Enterprise Performance Management System Configurator, see the *Oracle Hyperion Enterprise Performance Management System Installation and Configuration Guide* or Online Help.

➤ To select application information:

1. **In the Classic Application Wizard, click Create Application.**

   See "Setting Up Classic Applications" on page 249.

2. **Click Select.**

   See "About Updating Instances and Clusters" on page 46.

3. **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.

4. **Enter a description of the application.**

5. **Select a Shared Services application group.**

6. **Select an instance (cluster) to use for the application.**

   See "About Updating Instances and Clusters" on page 46.

7. **For Calculation Module, select the calculation module to use for the application, Business Rules or Calculation Manager.**

8. **For Application Type, select which type of application you are creating. For Planning applications, select General.**

9. **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 you select. During data entry, 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 315.

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 2010.

Table 62 Examples of Fiscal Year First Month and Fiscal Start Year Calendar Options

Fiscal Year First Month	Fiscal Start Year	Year - Period	Year Dimension
January	Same Calendar Year	Jan-10 to Dec-10	FY10
July	Same Calendar Year	Jul-10 to Jun-11	FY10
July	Previous Calendar Year	Jul-09 to Jun-10	FY10
February	Same Calendar Year	Feb-10 to Jan-11	FY10
February	Previous Calendar Year	Feb-09 to Jan-10	FY10
December	Same Calendar Year	Dec-10 to Nov-11	FY10
December	Previous Calendar Year	Dec-09 to Nov-10	FY10

➤ To set up the calendar:

1 **In the Calendar tab of the Classic Application Wizard, click Calendar.**

See "Setting Up Classic Applications" on page 249.

2 **Select a Base Period option to set how calendars roll up:**

- **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 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 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 2010 with a starting period of Jun creates the starting year as FY10, defined as Jul-09 to Jun-10.

For an application starting in January, only the **Same Calendar Year** option is available.

7 **If you set the base time period as 12 Months, select a Weekly Distribution option: Even, 445, 454, or 544.**

Weekly distribution sets the monthly distribution pattern, based on the number of fiscal weeks in a month. This 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 is considered to have five weeks, and the last two months in the quarter have four weeks.

8 **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. For information on Public Sector Planning and Budgeting, see the *Oracle Hyperion Public Sector Planning and Budgeting User's Guide*.

➤ To set up currencies:

1 **In the Classic Application Wizard, click Currencies.**

See "Setting Up Classic Applications" on page 249.

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.

For information on the plan type for Oracle Hyperion Public Sector Planning and Budgeting, Fusion Edition, see the *Oracle Hyperion Public Sector Planning and Budgeting User's Guide*.

➤ To select plan types:

1 **In the Classic Application Wizard, click Plan Types.**

See "Setting Up Classic Applications" on page 249.

2 **For each plan type in the application, select the Plan Type and specify a plan name.**

You must select least one Planning plan type. You can have up to three Planning plan types, and names can contain up to eight characters. (Although it is possible to enter more than eight bytes using single-byte and double-byte characters, an error message displays when the Essbase database is created.)

3 **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 Classic Application Wizard, click Finish.**

See "Setting Up Classic Applications" on page 249.

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 Finish.**

Registering Applications

You can use the Classic Application Wizard to reregister the Shared Services application group for an application.

➤ To register applications:

1 **Perform one action:**

- From Planning, select **Administration**, then **Application**, then **Register**.
- From EPM Workspace, select **Navigate**, then **Administer**, then **Classic Application Administration**, and then **Planning Administration**. Then select **Register Application**.

See "Setting Up Classic Applications" on page 249.

2 **Select an existing Planning application, and click Register.**

3 **Select a Shared Services application group, and click OK.**

Deleting Applications

You can use the Classic Application Wizard to delete Classic applications. Oracle recommends backing up the application before deleting. See "Backing Up Applications and Application Databases" on page 83.

➤ To delete applications:

1 Back up the application.

2 Perform one action:

   - From Planning, select **Administration**, then **Application**, then **Delete**.
   - From EPM Workspace, select **Navigate**, then **Administer**, then **Classic Application Administration**, and then **Planning Administration**. Then select **Delete Application**.

   See "Setting Up Classic Applications" on page 249.

3 Select an existing Classic application, and click **Delete Application.**

4 At the confirmation message, click **OK** if you want to proceed with the deletion.

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, Year, 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
- Japanese and English
- Korean and English
- Turkish and English

You can set alias tables to display members in applications. 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 **Explorer User Prompt**, 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 **Explorer User Prompt**, 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.**

Setting a Default Alias Table

If you create alias tables with aliases for Account, Currency, Entity, Scenario, Period, Version, Year, 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 **In Current Application Defaults, select Application Settings.**

3 **For Alias Table, select a default alias table.**

4 **Click Save (or Reset).**

Working with Dimensions

You can create and update dimensions by selecting **Administration**, then **Dimensions**.

Dimension Overview

Dimensions categorize data values. Seven dimensions are included with Planning: Account, Entity, Scenario, Version, Period, Year, 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. Essbase requires at least one dense dimension. 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 82 and "Setting Dimension Density and Order" on page 276.

About Dimension Hierarchies

Dimension hierarchies define structural and mathematical relationships, and consolidations between members in the database. Relationships are represented graphically in a collapsible hierarchy diagram. The levels below the database name are dimensions, and the levels below each dimension are members.

The Period dimension can contain the member YearTotal, which contains members Q1, Q2, Q3, and Q4. Members Q1, Q2, Q3, and Q4 contain their own members for the corresponding months in the year. To consolidate data values in the Period dimension, roll up monthly data values to get quarterly data values, and quarterly data values to get yearly data values.

Members of the same level that belong to the same dimension or member are called siblings. For example, Q1, Q2, Q3, and Q4 are siblings because they are at the same level in the hierarchy, and are members of the same member, YearTotal.

The members of a dimension are called children of the dimension. Members that belong to a member are called children of that member. The member YearTotal is a child of Period, the members of Q1, Q2, Q3, and Q4 are children of YearTotal, and Jan, Feb, and Mar are children of Q1. Q1 is the parent of Jan, Feb, and Mar, YearTotal is the parent of Q1, Q2, Q3, and Q4, and Period is the parent of YearTotal.

Working with Dimension Hierarchies

Task	Topic
Specify or change dimension properties.	Click **Edit**.
Add a dimension.	Click **Add Dimension**.
Search for a dimension member.	See "Finding Dimensions or Members" on page 266.
Expand or collapse the dimension hierarchy.	Click **Expand** or **Collapse**.

Task	Topic
Add or edit a dimension member.	Click **Add Child** or **Add Sibling**.
Move a dimension member.	See “Moving Members Within the Dimension Hierarchy” on page 266.
Delete a dimension member.	See “Deleting Members” on page 279.
Assign access to a dimension member.	See “Assigning Access to Members and Business Rules” on page 50.
View a member’s ancestors.	Click **Show Ancestors**.

Expanding and Collapsing Dimension Hierarchies

➤ To expand dimensions or members:

1 Select **Administration**, then **Manage**, then **Dimensions.**

2 For **Dimension**, select the dimension and member to expand.

3 Perform one action:

- Click **Expand**.
- Click ⊞.
- Click the closed folder.

➤ To collapse dimensions or members:

1 Select **Administration**, then **Manage**, then **Dimensions.**

2 For **Dimension**, select the dimension to collapse.

3 Perform one action:

- Click **Collapse**.
- Press the **Left Arrow**.
- Click ⊟.
- Click the open folders.

Navigating Dimension Hierarchies

- Press Up Arrow to move to the previous member.
- Press Down Arrow to move to the next member.
- In Page, enter the page to view and click Go or press Enter.
- Click Start, Prev, Next, or End to view other pages.

By default, 14 members are displayed per page. You can change this by setting preferences for Show the Specified Members on Each Dimensions Page.

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 Essbase 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 Essbase outline is generated with members in the order that is displayed.

Moving Members Within the Dimension Hierarchy

You can move one member or a group of members in the same branch. If you move Account members whose Valid For Plan Type settings differ from their new parent, the moved member settings change to match the setting of the new parents. If you move members whose Source Plan Type settings differ from their new parent, the moved member Source Plan Type is reset to match the first valid plan type.

➤ To move members or branches among siblings:

1. Select **Administration**, then **Manage**, then **Dimensions.**
2. For **Dimension**, select the dimension for the members to move.
3. Select the member or branch to move.
4. Perform one action:

- Click ⌃ to move the member up one position.
- Click ⌄ to move the member down one position.

➤ To move members, including parents and children:

1. Select **Administration**, then **Manage**, then **Dimensions.**
2. For **Dimension**, select the dimension with the members to move.
3. Select the member or branch to move.
4. Click **Cut.**

   You cannot Cut members after adding or editing dimensions, navigating to different pages, deleting members, or logging off 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.

Minimizing and Restoring Columns

➤ To minimize and restore columns:

1. Select **Administration**, then **Manage**, then **Dimensions.**
2. Select the column to minimize or restore.
3. Perform one action:
   - To minimize a column, double-click its heading, or right-click and select **Minimize**.
   - To return a column to its original width, double-click a minimized column heading, or right-click and select **Restore**.
   - To restore all minimized columns to original widths, right-click a column heading and select **Restore All**.

     Planning saves column widths for each dimension independently. Column widths are saved for the duration of the session when you add, edit, or delete a dimension member, or when you select another dimension.

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 Applications

➤ To view where members are used in applications:

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 63 Storage Options

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

About Dynamic Calc

With dynamically calculated members, Planning calculates data values of members, and disregards these values. The Essbase 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 data 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 multicurrency applications, 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 data forms in which label-only parents follow their first child member, as you cannot save data in the first child member. Instead, create data forms with label-only parents selected before their children, or do not select label-only parents for data 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 193).

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.

Data 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 multicurrency applications, 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 data 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 Essbase member formulas.

Examples of Using Account Types

Table 64 Using Account Types

Account Type	Purpose
Expense	Cost of doing business
Revenue	Source of income
Asset	Company resource
Liability and Equity	Residual interest or obligation to creditors
Saved assumption	Centralized planning assumptions ensuring consistency across the application

Summary of Account Types

Table 65 Summary of Account Types

Account Type	Time Balance	Variance Reporting
Revenue	Flow	Non-Expense
Expense	Flow	Expense
Asset	Balance	Non-Expense

Account Type	Time Balance	Variance Reporting
Liability	Balance	Non-Expense
Equity	Balance	Non-Expense
Saved Assumption	User-defined	User-defined

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

Time Balance Property

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

Table 66 Time Balance Properties

Time Balance Property	Description	Example
Flow	Aggregate of all values for a summary time period as a period total.	Jan: 10 Feb: 15 Mar: 20 Q1: 45
First	Beginning value in a summary time period as the period total.	Jan: 10 Feb: 15 Mar: 20 Q1: 10
Balance	Ending value in a summary time period as the period total.	Jan: 10 Feb: 15 Mar: 20 Q1: 20
Average	Average for all the child values in a summary time period as the period total.	Jan: 10 Feb: 15 Mar: 20 Q1: 15
Fill	The value set at the parent is filled into all its descendents. If a child value changes, the default aggregation logic applies up to its parent. Consolidation operators and member formulas overwrite Fill values when the members are recalculated.	Jan: 10 Feb: 10 Mar: 10 Q1: 10
Weighted Average - Actual_Actual	Weighted daily average, based on the actual number of days in a year; accounts for leap year, in which February has 29 days. In the example, the average for Q1 is calculated: (1) Multiply each month's value in Q1 by the number of days in the month, (2) Sum these values, (3) Divide the total by the number of days in Q1. Assuming it is a leap year, the result is calculated: (10 * 31 + 15 * 29 + 20 * 31) / 91 = 15	Jan: 10 Feb: 15 Mar: 20 Q1: 15
Weighted Average - Actual_365	Weighted daily average, based on 365 days in a year, assuming that February has 28 days; does not account for leap years. In the example, the average for Q1 is calculated: (1) Multiply each month's value in Q1 by the number of days in the month, (2) Sum these values, (3) Divide the total by the number of days in Q1. Assuming it is not a leap year, the result is calculated: (10 * 31 + 15 * 28 + 20 * 31) / 90 = 15	Jan: 10 Feb: 15 Mar: 20 Q1: 15

You can use the Weighted Average - Actual_Actual and Weighted Average - Actual_365 time balance properties only with a standard monthly calendar that rolls up to four quarters. For information on how Planning calculates and spreads data with the different Time Balance settings, see the *Oracle Hyperion Planning User's Online Help*.

Account Types and Variance Reporting

An account's variance reporting property determines whether it is treated as an expense when used in Essbase 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 67 Effect of Skip Options When Time Balance is Set to First

Skip Option	Description	Example
None	Zeros and #MISSING values are considered when calculating parent values (the default). In the example, the value of the first child (Jan) is 0, and zeros are considered when calculating the parent value, so Q1 = 0.	Jan: 0 Feb: 20 Mar: 25 Q1: 0
Missing	Excludes #MISSING values when calculating parent values. In the example, the value of the first child (Jan) is #MISSING, and #MISSING values are not considered when the calculating parent values, so Q1 = second child (Feb), or 20.	Jan: #MISSING Feb: 20 Mar: 25 Q1: 20
Zeros	Excludes zero values when calculating parent values. In the example, the value of the first child (Jan) is 0, and zero values are not considered when calculating parent values, so Q1 = the second child (Feb), or 20.	Jan: 0 Feb: 20 Mar: 25 Q1: 20
Missing and Zeros	Excludes #MISSING and zero values when calculating parent values. In the example, the value of the first child (Jan) is zero, and the value of the second child (Feb) is missing. Because missing and zero values are not considered when calculating parent values, Q1 = the third child (Mar), or 25.	Jan: 0 Feb: #MISSING Mar: 25 Q1: 25

Saved Assumptions

Use saved assumptions to centralize planning assumptions, identifying key business drivers and ensuring application consistency. You select time balance and variance reporting properties.

- Variance reporting determines the variance between budgeted and actual data, as an expense or non-expense.
- Time balance determines the ending value for summary time periods.

Examples of how time balance and variance reporting properties are used with saved assumption account members:

- Create a saved assumption of an expense type for variance reporting, assuming that the actual amount spent on headcount is less than the amount budgeted. To determine the variance, 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 data 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 68 Properties for User-Defined Custom Dimensions

Property	Value
Dimension	Enter a name that is unique across all dimensions.
Alias	**Optional**: Select an alias table. Enter a unique alternate name for the dimension.
Description	**Optional**: Enter a description.
Valid for Plan Types	Select plan types for which the dimension is valid. Clearing this option makes all members of the dimension invalid for the deselected plan type.
Apply Security	Allow security to be set on the dimension members; must be selected before assigning access rights to dimension members. Otherwise, dimensions have no security and users can access members without restriction.
Data Storage	Select a data storage option. The default is Never Share.

➤ To add or change user-defined dimensions:

1 Select **Administration**, then **Manage**, then **Dimensions.**

2 Click **Add Dimension.**

3 Set or change properties.

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 69 Dimension Properties

Property	Value
Dimension	Enter a dimension name.
Alias	**Optional**: Select an alias table and enter an alternate name of up to 80 characters. Follow the dimension naming restrictions.
Description	**Optional**: Enter a description.
Valid for Plan Types	Select plan types for which the dimension is valid. Not available for Entity or Account dimensions.
Apply Security	Allow security to be set on dimension members. If you do not select this option, there is no security on the dimension, and users can access its members without restriction. Must be selected before assigning access rights to dimension members.
Data Storage	Select data storage options.
Display Option	Set application default display options for the Member Selection dialog box. Select **Member Name** or **Alias** to display members or aliases. **Member Name:Alias** displays members on the left and aliases on the right. **Alias:Member Name** displays aliases on the left and members on the right.
Enable custom attribute display	Display available and selected attributes for dimensions with associated attributes. Enable custom attribute display for dimensions with attributes.

➤ To set properties for dimensions:

1 Select **Administration**, then **Manage**, then **Dimensions.**

2 Select the dimension.

3 Click **Edit.**

4 In **Dimension Properties**, set property options.

5 Click **Save** to save information to the relational database and see changes in the dimension hierarchy.

Click **Refresh** to revert to the previous values on the page and keep the page open.

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 264.

4 Set the order of precedence by selecting a dimension and clicking ⌃ or ⌄.

See "About Reordering Dimensions" on page 82.

Setting the Evaluation Order

The **Evaluation Order** tab enables you to specify which data type prevails when a data intersection has conflicting data types. For example, if Account members are set to the Currency data type, and Product members are set to the Smart List data type, you can set whether the Currency or Smart List data type prevails at an intersection.

➤ To set evaluation order:

1. Select **Administration**, then **Manage**, and then **Dimensions.**
2. Select **Evaluation Order.**
3. Select the plan type and click **Go.**
4. From **Available Dimensions**, select dimensions and move them to **Selected Dimensions**:
   - → moves selected dimensions
   - ⇉ moves all dimensions
   - ← removes selected dimensions
   - ⇇ removes all dimensions

   You need select only dimensions whose members have specific data types (that is, their data type is not “Unspecified”). The data type “Unspecified” does not conflict with another data type.
5. If you select multiple dimensions, set the order of precedence by clicking ⌃ or ⌄.
6. Click **Save.**

Working with Members

You can assign access rights to members, rearrange the dimension member hierarchy, and share members of the Entity, Account, and user-defined custom dimensions.

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 280.

Table 70 Member Properties

Property	Value
Name	Enter a name that is unique across all dimension members.
Description	**Optional**: Enter a description.
Alias	**Optional:** Select the alias table to store the alias name. Enter an alternate name for the member.

Property	Value
For Account members only: **Account Type**	Select the account type.
For Account members only: **Variance Reporting**	If the account type is Saved Assumptions, for Variance Reporting, select Expense or Non-Expense. Designate the saved assumption as a revenue, asset, liability, or equity account.
For Account members only: **Time Balance**	For Time Balance, select Flow or Balance.
For Account members only: **Exchange Rate Type**	For Exchange Rate Type, select an option.
For Account members only: **Data Type**	For Data Type, select a data type.
Distribution	Sets the weekly distribution. Available for leaf Account members if the option was selected when creating the application and the base time period is 12 months.
Data Storage	Select a data storage property used in building an Essbase cube. The default is Never Share for new custom dimension members (except root members).
Two Pass Calculation	Recalculate values of members based on values of parent members or other members. Available for Account and Entity members with Dynamic Calc or Dynamic Calc and Store properties.
For Entity members only: **Base Currency**	Select the Entity member's base currency.
Display Option	Set application default display options for the Member Selection dialog box. Select **Member Name** or **Alias** to display members or aliases. **Member Name:Alias** displays members on the left and aliases on the right. **Alias:Member Name** displays aliases on the left and members on the right.
Plan Type	Select plan types for which the member is valid. 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. **Caution!** Deselecting a plan type for dimension members after data is entered into applications may result in loss of data when applications are refreshed. For account members, data is lost if the deselected plan type is the source plan type.
For Entity members only: Base Currency	Select the base currency for the Entity member.
For Account members only: Source Plan Type	Select the source plan type for the member. A shared member is a pointer to the base member and is not stored in Essbase; 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.
Smart Lists	**Optional**: Select a Smart List to associate with the member.

➤ To add or edit members:

1 **Select Administration, then Manage, 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**.

4 On **Member Properties**, set or change member properties.

If you do not see the new member on the page, click **Next**.

5 Click **Save** to save information to the relational database and see changes in the dimension hierarchy.

6 Refresh the database so edited members are visible to planners entering data.

7 After creating a dimension member, you typically complete these tasks:

- Assign access. See "Assigning Access to Members and Business Rules" on page 50.
- Specify attributes.

Deleting Members

Caution! 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 applications. Deleting entity members deletes all planning units (including data) associated with them. Before beginning this procedure, perform a backup. See "Backing Up Applications and Application Databases" on page 83 and the *Oracle Hyperion Enterprise Performance Management System Backup and Recovery Guide*.

Before deleting members, understand where in the application they are used (in which data 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 data form, you must delete it from the data 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 193.

➤ 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

Caution! 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. Before beginning this procedure, perform a backup. See "Backing Up Applications and Application Databases" on page 83 and the *Oracle Hyperion Enterprise Performance Management System Backup and Recovery Guide*.

➤ To delete a parent member and all its descendants from the dimension hierarchy:

1 Select **Administration**, then **Manage**, then **Dimensions**.

2 Select the dimension whose member and descendants to delete.

3 Select the member whose branch to delete.

4 Click **Delete**.

5 Click **OK**.

Viewing Member Properties from Data Forms

➤ To view member properties from data forms:

1 In the data 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 Planning applications. 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 Business Rules and Smart View. For Oracle's Hyperion® Business Rules, you might not be able to select shared members for actions such as Copy Data.

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 by Essbase.

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 282. 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 282. 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, Essbase compares characters. 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 Online Help*.

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 data forms. Data values for attribute values are dynamically calculated but not stored.

Creating Attribute Values

You can define attribute values for sparse dimensions, which are typically the Entity and user-defined custom dimensions. After you define an attribute value for a dimension, you can assign it to members of that dimension.

➤ To create attribute values:

1. Select **Administration**, then **Manage**, then **Dimensions.**
2. Select the sparse dimension for which to create an attribute value.
3. Select the top level in the dimension hierarchy.
4. Click **Custom Attributes.**
5. On the **Manage Attributes and Values** page, select the attribute for which to specify a value.
6. Above the **Attribute Values** column, click **Create.** If the options are available, you can click **Add Child** or **Add Sibling.**
7. On **Create Attribute Value**, in **Name**, enter a name.
8. Press **Enter** or **Save.**
9. Click **Cancel.**

Assigning Attribute Values to Members

You can assign attribute values members of a dimension that are defined as sparse for all plan types. Attribute values must be assigned to the same-level sparse dimension members. Otherwise, errors display during refresh.

➤ To assign attribute values to members:

1. Select **Administration**, then **Manage**, then **Dimensions.**
2. Select the sparse dimension for whose member you want to assign an attribute value.
3. In the **Dimension** hierarchy, select a member to which to assign an attribute value.
4. Click **Edit.**

   For members assigned attribute values: Click **View** to change a member's attribute value.
5. Select **Attribute Values.**
6. Select attribute values to assign to the member.
7. Perform an action:
   - To assign the value to the selected member, click ⇥.
   - To remove a value from the selected member, select the value to remove and click ⇤.
   - To remove all values from the selected member, click ⇇.
8. Click **Save.**

Modifying Attribute Values

➤ To modify attribute values:

1. Select **Administration**, then **Manage**, then **Dimensions.**
2. Select the sparse dimension for which to modify an attribute value.
3. Select the top level in the dimension hierarchy.
4. Click **Custom Attributes.**
5. For **Attributes**, select the attribute containing the value to modify.
6. For **Attribute Values**, select the attribute value.
7. Above **Attribute Values**, click **Modify.**
8. On **Modify Attribute Value**, in **Name**, enter a name.
9. Click **Save.**

Deleting Attribute Values

When you delete an attribute value, it is removed from custom dimension members to which it is assigned.

➤ To delete attribute values:

1 Select **Administration**, then **Manage**, then **Dimensions.**

2 Select the sparse dimension containing the attribute for which to delete a value.

3 Select the top level in the dimension hierarchy.

4 Click **Custom Attributes.**

5 For **Attributes**, select the attribute containing attribute values to delete.

6 For **Attribute Values**, select attribute values to delete.

To select all attribute values for deletion, select **Attribute Values**.

7 Above the **Attribute Values** column, click **Delete.**

8 Click **OK.**

9 Update and validate business rules and reports.

Customizing Calendars

Use the Period dimension to work with the yearly calendar rollup structure. When creating the application, the administrator specifies the base time periods that span the Planning database. Use the Year dimension to add years to the calendar.

Table 71 Calendar Tasks

Task	See Topic
Define how years roll up.	"Defining How Calendars Roll Up" on page 285.
Create and edit Summary Time Periods.	"Creating and Editing Summary Time Periods" on page 286.
Delete Summary Time Periods.	"Deleting Summary Time Periods" on page 286.
Work with the fiscal year and period hierarchy.	"Working with Years" on page 287.

Defining How Calendars Roll Up

Table 72 Calendar Roll Up

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

After the application calendar is created, you cannot change the base time period or reduce the number of years in the calendar. Administrators can change the names, descriptions, aliases, and ranges of the summary time periods in the hierarchy.

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 Years

Use the Year dimension to work with calendar years. You can add years, change the current year or period, and assign aliases to time periods or years.

Table 73 Year Tasks

Task	See Topic
Add years to the calendar.	"Adding Years to the Calendar" on page 287.
Add or update the description and alias for a year.	"Editing Year Information" on page 288.
Set the fiscal year and work with years.	"Setting the Fiscal Year" on page 288.

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 **Year.**
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.**

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 **Year** or **Period.**

3 Click **Options.**

4 On **Set Current Time 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 **Save.**

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 **Year.**

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 applications use, including for reporting
- How currencies display in reports and data forms
- How currencies convert to other currencies
- Whether a triangulation currency converts currencies
- When currency conversions occur

Enabling Multiple Currencies

If applications support multiple currencies, you can enable multiple currencies per entity on data forms. See “Setting Row and Column Layout” on page 125. When selecting business rules for data forms, you can select the Calculate Currencies business rule to convert values among the available currencies. See “Selecting Business Rules” on page 140.

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 data 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.
- Applications 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 Data Form.
- Currency codes associated with input values are stored as numeric values in Essbase. 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 in Essbase (which is not displayed on data 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 data 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 Oracle Hyperion Smart View for Office, Fusion Edition. 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 data form. It converts data from the local currency to the reporting currency specified on the data form, applying the exchange rate conversions. It:

- Does not calculate subtotals. To subtotal values, run the Calculate Data 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 data forms during data 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 data form with the Local member selected for the Currency dimension. When you select Yen as the currency member for the data 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 data 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 applications use 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. **Perform one action:**
   - 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.
     - Optional: For **Thousands Separator**, select how to display the thousands separator (it must differ from the decimal separator).
     - Optional: For **Decimal Separator**, select how to display numbers with decimal values (it must differ from the thousands separator).
     - Optional: For **Negative Sign**, select how to display negative numbers:
       - **Prefixed minus**: -1000.
       - **Suffixed minus**: 1000-
       - **Parentheses**: (1000)
     - Optional: For **Negative Color**, select the display color.
     - Optional: Select **Reporting Currency** to set the currency as a reporting currency.
5. **Click Save.**

Editing Currencies

➤ To edit currencies:

1 Select **Administration**, then **Manage**, then **Dimensions.**

2 Select **Currency.**

3 Select the currency to edit.

4 Click **Edit.**

5 Modify properties:

- To change the currency's symbol, for **Symbol**, enter or select the symbol.
- For **Description**, enter a descriptive name.
- For **Scale**, set how to enter and display the currency.
- For **Triangulation Currency**, select the currency to use as the common third currency for conversion.
  - Optional: For **Thousands Separator**, select how to display the thousands separator (it must differ from the decimal separator).
  - Optional: For **Decimal Separator**, select how to display numbers with decimal values (it must differ from the thousands separator).
  - Optional: For **Negative Sign**, select how to display negative numbers:
    - **Prefixed minus**: -1000.
    - **Suffixed minus**: 1000-
    - **Parentheses**: (1000)
  - Optional: For **Negative Color**, select the display color.
  - Optional: Select **Reporting Currency** to set the currency as a reporting currency.
- For **Alias Table**, select an alias table and modify the currency alias.
- Select or clear **Reporting Currency**.
- You cannot modify the Code property.

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 data form having values for the Japan entity and the data 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 applications.

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 75.

Hsp_Rates Dimension

Multiple-currency applications include 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 Dense. You can change this (see "Setting Dimension Density and Order" on page 276).

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.

Task	Topic
Create a scenario.	See "Creating Scenarios" on page 297.
Edit a scenario.	See "Editing Scenarios" on page 298.
Delete a scenario.	See "Deleting Scenarios" on page 298.
Copy a scenario.	See "Copying Scenarios" on page 299.
Delete supporting detail associated with a scenario.	See "Deleting Supporting Detail Associated With a Scenario" on page 242.

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 data 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 Rights

Specify access rights to members of the Scenario dimension for groups or users. Access rights determine users or groups can view or modify data. A user or group can have only one of these access rights: Read, Write, or None. Access rights for a user can be combined based on groups to which the user belongs.

Creating Scenarios

➤ To create scenarios:

1. Select **Administration**, then **Manage**, 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 data form. You must first remove references to scenarios from data forms and assign different scenarios.

➤ To delete scenarios:

1 Select **Administration**, then **Manage**, then **Dimensions**.

2 Select **Scenarios**.

3 Select the scenarios to delete. At least one scenario must remain in the application.

4 Click **Delete**.

5 Click **OK**.

6 Update and validate business rules and reports.

Copying Scenarios

Only scenario properties are copied. Data values and access rights associated with the original scenario are not copied to the new scenario.

➤ To copy scenarios:

1 Select **Administration**, then **Manage**, then **Dimensions.**

2 Select **Scenarios.**

3 Select the scenario to copy.

4 Click **Copy Scenario.**

5 For **Copy to Scenario**, enter a name.

6 **Optional**: For **Description**, enter a description.

7 For **Start Yr.**, **Start Period**, **End Yr.**, and **End Period**, select the time period to associate with the scenario.

8 **Optional**: For **Exchange Rate Table**, select an exchange rate table to associate with the scenario.

If an application uses multiple currencies, associate a scenario with an exchange rate table to enable currency conversion.

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

10 **Optional**: Select **Include BegBal as Time Period** to include the BegBalance time period in this scenario for currency conversion.

11 **Optional**: Select **Enabled for Process Management** to include this scenario in approvals.

12 Click **Save.**

Specifying Versions

Use versions to group data used by applications.

Task	**Topic**
Create a version.	See "Creating Versions" on page 300.
Edit a version.	See "Editing a Version" on page 300.
Delete a version.	See "Deleting Versions" on page 301.

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.**

5 **Optional**: For **Version**, modify the name of the version.

6 **Optional**: For **Typelist**, modify the type of version to display for the application:

- **Standard Target** - Values are entered from the parent level down.
- **Standard Bottom Up** - Values are entered at the lowest member level and aggregated upward.

7 **Optional**: For **Description**, enter a description.

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

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

This option is not available for target versions.

10 Click **Save.**

11 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 data forms. You must remove references to versions from data forms and assign another version to axes. At least one version must remain in the application.

➤ To delete versions:

1 Select **Administration**, then **Manage**, then **Dimensions.**

2 Select **Version.**

3 Select the versions to delete.

4 Click **Delete.**

5 Click **OK.**

If planning units are started, you cannot delete versions associated with them.

6 Update and validate business rules and reports.

Displaying Versions

➤ To display versions:

1 Select **Administration**, then **Manage**, then **Dimensions.**

2 Select **Version**.

3 For **Display**, select the version types to display.

- **Standard Target**: Values are entered from the parent level down.
- **Standard Bottom Up**: Values are entered at the lowest member level and aggregated up.

Sorting Version and Scenario Members

You can sort version and scenario members in ascending or descending order. Sorting members affects the Essbase 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 Essbase 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 Essbase outline is generated with members placed in the same order as displayed on the Dimensions tab.

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.

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 83. If you are using the Y-T-D or P-T-D member, you must rename the Year or Period dimension so it does not conflict with the reserved Dynamic Time Series generation names, Year and Period. Before using Y-T-D, rename the Year 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.

It is recommended 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 Online Help*.

➤ 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).

5 **Optional: Select an alias table and type an alias name. (If necessary, resize the window to view the fields.)**

See "Working with Alias Tables" on page 261.

6 **Click Save.**

Additional Supported Essbase Features

Additional Essbase features are now 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 264). You can also use shared members for these dimensions, and can set two pass calculations at the root level for all dimensions. For example, you can:

Feature	More Information
In Scenario and Version dimensions, create hierarchies and use shared members. If you assign children to bottom-up versions, these versions display as read-only parents on data forms.	See "Setting Up Scenarios" on page 296 and "Specifying Versions" on page 299.
In the Period dimension, create alternate hierarchies and use shared descendants. Data Storage for all time periods can be set to any valid Data Storage value. The Consolidation operator for all time periods, including BegBalance, can be set to any valid consolidation operator. For example, it can be set to + instead of ~ (ignore).	See "Working with Years" on page 287, "Editing the BegBalance Member" on page 289, and "Editing Exchange Rate Tables" on page 79.
Turn on two pass calculation at the root level, for example, for Account. **Caution!** 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.	See "Adding or Editing Members" on page 277.
For attributes, create hierarchies and assign aliases.	See "Working with Attributes" on page 281.

Note: For multicurrency applications, the pre-built currency conversion calc scripts do not function properly if you change data storage to dynamic for any Scenario, Version, Period, or Year 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 Classic Applications

You can create and update Classic applications if you are assigned the Shared Services Dimension Editor and Planning Application Creator roles. For detailed information, see the *Oracle Hyperion Enterprise Performance Management System User and Role Security 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 83. 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 32.

➤ To refresh the application database using a utility:

1. **Back up the application. See "Backing Up Applications and Application Databases" on page 83.**
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 45.
3. **Launch `CubeRefresh` from the `planning1` directory by entering the command and its parameters in the command line:**

   Windows: `CubeRefresh.cmd [-f:`*`passwordFile`*`] /A:`*`application_name`*` /U:`*`user_name`*` [/C|/R] /D [/F[S|V]][/RMIPORT:`*`rmi_port`*`] [/L] [/DEBUG]`

   UNIX: `CubeRefresh.sh [-f:`*`passwordFile`*`] /A:`*`application_name`*` /U:`*`user_name`*` [/C|/R] /D [/F[S|V]][/RMIPORT:`*`rmi_port`*`] [/L] [/DEBUG]`

   - **Optional:** If an encrypted password file is set up, you can specify `[-f:`*`passwordFile`*`]` as the first parameter in the command line, where *passwordFile* is the full file path and name for the password file. See "Suppressing Password Prompts in Planning Utilities" on page 44.

- **application_name**: The name of the Planning application on which to run the create or refresh.
- **user_name**: The administrative user who has rights to create or refresh the application.
- **/C** or **/R**: The function to perform on the application:
    - **/C**: Create the database outline.
    - **/R**: Refresh the database outline.
- **/D**: Specify the database during create or refresh.
- **/F**: Use security filters for all users of the application. Optionally, use with **S**, **V**, or **SV**:
    - **/FS**: Generate shared member security filters.
    - **/FV**: Validate security filters, but do not save the filters.
    - **/FSV**: Validate the shared member security filters.
- **/RMIPORT**: Specify an RMI port number different from the preset value of 11333.
- **/-L**: Default option, to create or refresh by connecting to an application server on a local or remote machine. (You can use **/L** to create or refresh without connecting to an application server, for example, if the application server is not running.)
- **/DEBUG**: Specify detailed error messages.

4 If prompted, enter your password.

5 View the result of the application refresh, including errors. Completion status is shown on the console.

This example shows a command line used to refresh a database for all users logged on to the specified application. It uses the default RMI port and includes shared member security filters:

Windows:`CubeRefresh.cmd /A:app1 /U:admin /R /D /FS`

UNIX:`CubeRefresh.sh /A:app1 /U:admin /R /D /FS`

Refreshing Databases with Level 0 Members Set to Dynamic Calc

If a level 0 member in the database outline is set to Dynamic Calc or Dynamic Calc and Store, 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, Year, 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 45.

Table 74 SortMember Utility Parameters

Parameter	Description
`[-f:passwordFile]`	**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 44.
servername	The server name where the Planning application resides
username	The name of the Planning administrator
application	The name of the Planning application containing the dimension members to sort
member	The parent member whose children or descendants to sort
`children|descendants`	Whether to sort by children or descendants: sorting by children affects only members in the level immediately below the specified member; sorting by descendants affects all descendants of the specified member
`ascend|descend`	Whether to sort by ascending order or by descending order

For example:

```
SortMember localhost admin BUDGET account200 descendants ascend
```

2 **If prompted, enter your password.**

If the application server, HBR Service, 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, HBR Service, 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 45.

Table 75 DeleteSharedDescendants Utility Parameters

Variable	Description
`[-f:`*`passwordFile`*`]`	**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 44.
`servername`	The server name on which the Planning application resides.
`username`	The name of the Planning administrator
`application`	The name of the Planning application containing the shared dimension members to delete.
`member`	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.

For example:

```
DeleteSharedDescendants localhost admin BUDGET account200
```

2 **If prompted, enter your password.**

3 **To see the results of running the utility, check the log files that are generated in the *`EPM_ORACLE_INSTANCE`*`/diagnostics/logs/planning` directory:**

- `DeleteSharedDescendants.log`: contains status messages.

- `DeleteSharedDescendantsExceptions.log`: contains error messages.

If you run this utility while the Manage Database task is open, no error message is displayed, but the member is not deleted. The LOG and CMD files in the *`EPM_ORACLE_INSTANCE`*`/diagnostics/logs/planning` directory show that one shared member was found, but 0 shared members were deleted.

Using Smart Lists, UDAs, and Member Formulas

For Classic applications, 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 data 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:
  - Turn Smart Lists on or off for data forms.
  - Use Smart List values in member formulas.
  - Set how #MISSING cells associated with Smart Lists display in data 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:** Synchronize Smart Lists between a Classic Planning application and reporting application by clicking **Synchronize**. See "Synchronizing Smart Lists in Reporting Applications" on page 310.

Synchronizing Smart Lists in Reporting Applications

For Classic Planning applications 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 Planning Applications to Reporting Applications" on page 235.

➤ 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 311.

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.

➤ 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 76 Smart List Properties

Property	Description
Smart List	Enter a unique name containing only alphanumeric and underscore characters (for example: Position) and no special characters or spaces. Smart List names can be referenced in formula expressions.
Label	Enter the text to display when the Smart List is selected. Spaces and special characters are allowed. Smart List labels can reference a resource, which can be translated into different languages. See "About Customizing Text, Color, and Images" on page 326.
Display Order	How Smart Lists are sorted in the drop-down list: by ID, Name, or Label
#MISSING Drop-Down Label	Enter a label (for example, "No Justification") to be displayed as an entry in the Smart List whose value is #MISSING. Notes: ● It displays as the first selection in the Smart List drop-down, allowing #MISSING as a selection in the data 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 data form. ● #MISSING labels determine only the display of cells with #MISSING data; #MISSING remains the stored value.
#MISSING Data 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. ● **Data Form Setting**: Displays #MISSING or leaves cells blank, depending on the Display Missing Values As Blank selection for the data 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 77 Smart List Entries

Entry Property	Description
ID	Unique number that sets the order for the displayed entry. Customizable only if **Automatically Generate ID** is not selected on the Properties tab.
Name	Unique alphanumeric name containing alphanumeric and underscore characters (for example: Customer_Feedback) and no special characters or spaces
Label	Displayed text for the Smart List entry on the drop-down list (for example: Customer Feedback).

Items highlighted in red are duplicates.

4 **Perform one action:**

- Click **Save**.
- Select **Preview**.

Previewing Smart Lists

Preview the defined Smart List on the Preview tab. The tab shows the Smart List as displayed in a drop-down list or a table.

Displaying #MISSING with Smart Lists

Administrators set values displayed in Smart Lists and data cells, including the display when no data is in the cell. Cells can display no value, #MISSING, or (for cells associated with Smart Lists) a specified value.

Use these options to control the display of #MISSING when cells are not in focus:

Option	Guideline
Blank	When designing data forms, select **Display Missing Values as Blank**. When setting Smart List properties, select **Data Form Setting**.
#MISSING	When designing data forms, do not select **Display Missing Values as Blank**. When setting Smart List properties, select **Data Form Setting**.
A custom label, such as “No Change”	When setting Smart List properties, enter the custom label in the **#MISSING Drop-Down Label** field (for example, `No Change`). Select **Drop-Down Setting**.

Working with UDAs

You can use user-defined attributes (UDAs), descriptive words or phrases, within calc scripts, member formulas, and reports. UDAs return lists of members associated with the UDA. For example:

- You can use the `HSP_UDF` UDA to prevent a formula from being overwritten when the application is refreshed. You must log on to each database associated with the Planning application and create the member formula with a UDA. The syntax for this UDA is: `(UDAs: HSP_UDF)`.

- If you use the `@XREF` function to look up a data value in another database to calculate a value from the current database, you can add the `HSP_NOLINK` UDA to members to prevent the `@XREF` function from being created on all plan types that are not the source plan type selected for that member.
- For a Product dimension with several product members, you can create a UDA called New Products and assign this UDA to the new products in the Product dimension hierarchy. Then you can base certain calculations on the designation New Products.
- For the budget review process, rather than creating a data validation rule for each owner in a product line (some product lines can have hundreds of owners), you can create a UDA containing the user names that apply to members using the planning unit hierarchy. Then in the data validation rule, you can enter a lookup function which will return the user names stored in the UDA for the current member. For example, create a UDA for each user in the promotional path and assign a prefix to the UDA name (for example, ProdMgr:Kim).

For more information on creating and using UDAs, see "Considerations for Working with Essbase" on page 32 and *Oracle Essbase Database Administrator's Online Help*.

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 **Add.**
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 Add.**

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 Essbase formulas (see the *Oracle Essbase Database Administrator's Guide*).
- Predefined Planning 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 Member Formula.**

5 **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.

6 **Optional: To check the validity of the member formula, click Validate Member Formula and OK.**

7 **Click Save.**

Before you click Save, clicking Refresh restores the previous member formula information.

Viewing Details of Formula Validation

➤ To view details of the member formula validation:

1 **On Member Formulas, click Validate Member Formula.**

2 **If the member formula is not valid, click Show Details.**

If the member formula is valid, **Show Details** is not selectable.

3 **Click OK.**

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 Member Formula 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 **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.

6 **Optional:** To check the validity of the member formula, click **Validate Member Formula** and **OK**.

7 **Optional:** If there are errors in the member formula, click **Show Details** to view a description.

8 **Optional:** Click **Refresh** to restore the previous member formula if you do not want to save the changes you made to the member formula.

9 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 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, Essbase replaces Status.Departed with 2 in the member formula (Planning treats Smart Lists as numbers). If the Smart List ID for Departed is 2, Essbase puts 2 in the calculation and stores 2 in the database.

Write Smart Lists in this format: [SmartListName.SmartListEntry]

Formula Expressions

Planning formula expressions can include these predefined variables and functions.

Table 78 Variables in Formula Expressions

Variable	Description
OpenInputValueBlock	Generates an IF statement if the Planning application is a multicurrency application, or an empty string if it is one currency application. Used with ClosedInputValueBlock.
CloseInputValueBlock	Generates an End IF statement if the Planning application is a multicurrency application, or an empty string if it is one currency application. Used with OpenInputValueBlock.
NumberOfPeriodsInYear	Returns the number of time periods in the year
NumberOfYears	Returns the number of years in the application

Table 79 Functions in Formula Expressions

Function	Description
Dimension(dimTag)	Returns the name of a predefined dimension. The dimtags are: ● DIM_NAME_PERIOD ● DIM_NAME_YEAR] ● DIM_NAME_ACCOUNT ● DIM_NAME_ENTITY ● DIM_NAME_SCENARIO ● DIM_NAME_VERSION ● DIM_NAME_CURRENCY
Period(periodName)	Returns the specified period. The periodName options are: ● FIRST_QTR_PERIOD ● SECOND_QTR_PERIOD ● THIRD_QTR_PERIOD ● FOURTH_QTR_PERIOD ● FIRST_PERIOD ● LAST_PERIOD
CrossRef(accountName)	Generates a cross reference for the account
CrossRef(accountName, prefix)	Generates a cross-reference for the account. The account name contains a prefix that you define. The default prefix is No, followed by a blank space and the account name, for example, No Salary.
getCalendarTPIndex()	Generates a member formula that returns an index for the time period; the index is based on the calendar year.
getFiscalTPIndex()	Generates a member formula that returns an index for the time period; the index is based on the fiscal year.
CYTD(memberName)	Generates a calendar year-to-date formula for the member
CYTD(memberName, calTpIndexName, fiscalTpIndexName)	Generates a calendar year-to-date formula for the member, and the time period index based on the calendar year and fiscal year. Use when members are renamed. The default member names are “Cal TP-Index” and “Fiscal TP-Index.”

Understanding Common Errors

Follow the rules of syntax carefully. If formula expression syntax contains errors, error messages are returned after you validate the member formula. To get information about error messages, click Show Details on the Member Formula tab. The most common error message is “Failed to execute.” This occurs when you use parameters in the expression incorrectly. These actions cause “Failed to execute” error messages:

- Entering the wrong number of parameters in the formula expression
- Misspelling member names, functions, or variable names
- Not surrounding member names with quotation marks
- Including numbers where strings are required

13 Customizing Planning Web Client

In This Chapter

Customizing Reports

Planning includes templates that control the layout and content of PDF reports of data forms, data form definitions, task lists, and planning units. You can use the templates as is. You can also customize the templates to add company logos, and special formatting for features such as shading, page size, orientation, font, font size, headers, percentage of page used for headers, number of data columns per page, and precision.

To customize reports, you must install and configure Microsoft Office Word 2000 or later and Oracle Business Intelligence Publisher Desktop. You can then use Word's BI Publisher menu to update information in the report using a sample .XML file. You can also use Word features to customize formatting. You make the template available by saving the .RTF file with the appropriate name, and placing it in the classpath or in the `HspJS.jar` file.

You can customize four report types, using the corresponding sample and template files. For information on creating the reports, see the related topics.

Report Type	Sample File Name	Template Name	Related Topics
Data Form	`PlanningFormSample.xml`	`PlanningFormTemplate.rtf`	See "Creating Simple Data Forms" on page 124.
Data Form Definition	`PlanningFormDefSample.xml`	`PlanningFormDefTemplate.rtf`	See "Printing Data Form Definitions" on page 150.

Report Type	Sample File Name	Template Name	Related Topics
Task List	`PlanningTaskListSample.xml`	`PlanningTaskListTemplate.rtf`	See the *Oracle Hyperion Planning User's Online Help*.
Planning Unit Annotations	`PlanningPUAnnotationSample.xml`	`PlanningPUAnnotationTemplate.rtf`	See "Printing Planning Unit Annotations" on page 197.

This topic gives general customization instructions. For detailed procedures, see the documentation installed with Word and BI Publisher. The BI Publisher install guide and user's guide are also available here:

`http://download.oracle.com/docs/cd/E10091_01/welcome.html`

➤ To install BI Publisher Desktop:

1. **Download the most recent version of BI Publisher Desktop from this location:**

   `http://www.oracle.com/technology/software/products/publishing/index.html`

   To customize reports, you only need to install BI Publisher Desktop. BI Publisher is not required.

2. **Save the zip file to your drive, and extract it with Use Folder Names selected.**
3. **Close all Microsoft Office applications.**
4. **Navigate to the directory where you extracted the zip file, and double-click the `setup.exe` application.**
5. **Follow the instructions in the BI Publisher installation wizard to perform a basic installation, accepting the default settings.**

➤ To customize reports:

1. **Open the Planning `HspJS.jar` file, installed by default in `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.

   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`.

Type of Template	File Name
All data forms	`PlanningFormTemplate_Custom.rtf`
Specific data form	`PlanningFormTemplate_`*`Data_Form_Name`* For example, to apply the template to a data form called Income, save the template as `PlanningFormTemplate_Income.rtf`.
Data form definition	`PlanningFormDefTemplate_Custom.rtf`
Task list status reports	`PlanningTaskListTemplate_Custom.rtf`
Planning unit annotation reports	`PlanningPUAnnotationTemplate_Custom.rtf`

You must save the template file in a location on the classpath so it is available to the Web application server. Insert the template file in the `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 links from the Tools menu and click links to open pages in secondary windows in their browser. See "Specifying Custom Tools" on page 233.

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 323.

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 80 Major Sections in the global.css File

Customizable Sections	Sections Not Recommended for Customizing
Non-Navigation Styles	Tadpole Menubar Tadpole Menuitem Menu Styles (order is important)
Tadpole Masthead	Tadpole Minimized Masthead
Tadpole Content Area	Tadpole Toolbar
Tadpole Logon Styles	Tadpole Viewpane
Tabs Tab Anchor Tags Tab Body	Tadpole Tree
TaskTabs Task Tab Body	
Groupbox Groupbox Anchor Tabs	

Customizable Sections	Sections Not Recommended for Customizing
Buttons When Drawing Buttons in HTML When Using Button Tag When Using Input Tag	

➤ To customize a cascading style sheet:

1 **Extract the `HyperionPlanning.ear` file and then the `HyperionPlanning.war` file to a temporary location.**

See "Location of Cascading Style Sheet Files" on page 323.

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 data forms to reflect members' hierarchical positions within the row or column axis, see "Customizing the Style of Row and Column Members in Data Forms" on page 325.

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

This section provides topics on common customizations:

- "Including Your Logo" on page 324
- "Changing the Color of the Content Area Background" on page 325
- "Changing the Color of Hyperlinks and Visited Hyperlinks" on page 325

Including Your Logo

You can update the `global.css` file to use your organization's logo or image.

➤ To include your logo:

1 **Place the image in this directory:**

`WebAppServer/ApplicationDirectory``/ui_themes/tadpole/images_global`

2 **Open the `global.css` file using a style sheet editor tool.**

3 **In the Tadpole Masthead section of the `global.css` file, change the reference to the image:**

```
.masthead TD.co_brand {

background-image: URL("images_global/image_name.gif");
```

4 **Open the `BannerTop.jin` file, and make the same change to the `<td>` tag that has this comment: `Insert Co-Brand Logo onto this TD`.**

```
<!--Insert Co-Brand Logo onto this TD -->

<td class="co_brand logo_padding">&nbsp;


<img class="co_brand" src="<%=iconPath %>spacer.gif"/>

</td>
```

5 **To change the image on the Logon screen to your organization's logo or image, replace the `logon_panel.jpg` file with the image in this directory:**

`WebAppServer/ApplicationDirectory``/ui_themes/tadpole/images_global`

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 81 Changing the Color of the Content Area Background

Default Content Area Background	White Content Area Background

`.content table.content {`
`background: #`**`e5eaef`**`;`
`}`
	`.content table.content {`
	`background: #`**`ffffff`**`;`
	`}`

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 82 Changing the Color of Hyperlinks and Visited Hyperlinks

Black Hyperlinks and Visited Hyperlinks	Red Hyperlinks and Visited Hyperlinks

`a:link, a:visited {`
`color: #`**`000000`**`;`
`}`
	`.content table.content {`
	`background: #`**`ff0000`**`;`
	`}`

Customizing the Style of Row and Column Members in Data Forms

You can customize Planning cascading style sheets to display data forms with different styles for the levels of members in rows and columns, depending on their hierarchical position in the data form. Note:

- Customized styles are applied regardless of the dimension displayed in data 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 data 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 322.

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

Table 83 Header Tags in Planning.css File

Rows	Columns
rowHeader_0	columnHeader_0
rowHeader_1	columnHeader_1
rowHeader_2	columnHeader_2
rowHeader_3	columnHeader_3
rowHeader_4	columnHeader_4

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.

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 326
- "Customizing Text in Non-Latin Languages" on page 328
- "Restricted Characters in Customized Messages" on page 329

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 329.
- 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 322.
- 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 and then perform a redeploy for the changes to be picked up.**

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 148.

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.

➤ 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 84 Examples of Language Encoding Parameters

Language	Encoding Parameter
Russian	Cp1251
Turkish	Cp1254
Japanese	SJIS
Simplified Chinese	EUC_CN
Traditional Chinese	Big5
Korean	EUC_KR

a. Ensure that the `native2ascii` program included with the Sun JDK is installed, and note the installation path to this program. Some application servers install this program for you. If it is not installed, you can install the JDK.

b. Open a command prompt.

c. Type the path to the `native2ascii` program, followed by the language encoding parameter and the names of the source and target files to be created. For example:

```
C:\j2sdk1.4.2_15\bin\native2ascii —encoding Cp1251
HspCustomMsgs_ru.source HspCustomMsgs_ru.properties
```

6 **Extract the `HyperionPlanning.ear` file to a temporary location.**

7 **Copy the properties file (such as `HspCustomMsgs_ru.properties`) to the `custom` directory where `HyperionPlanning.war` was extracted.**

8 **Re-create `HyperionPlanning.ear` and place the ear file in the following location:**
EPM_ORACLE_INSTANCE_HOME`\products\Planning\AppServer\InstallableApps\Common`

9 **Remove the extracted `\HyperionPlanning` from the temporary location.**

10 **Restart Planning.**

11 **Restart the Web application server.**

Restricted Characters in Customized Messages

You can modify text strings to meet your business needs. Avoid using characters that can cause errors in custom messages, depending on how the text string is used and whether the string is generated by Java or JavaScript. (In most cases, text messages are generated by Java, and pop-up boxes and some buttons are generated by JavaScript.) For example, errors occur if you add this string inside a JavaScript call because JavaScript cannot parse double quotation marks:

```
someJavaScript("<%= HspMsgs.SOME_RESOURCE %>");
```

The best practice is to avoid using these characters in custom messages:

- Single quotation marks
- Double quotation marks
- Back slashes
- Forward slashes

Customizing Colors, States, and Actions for Process Status

These procedures require knowledge of how to maintain and manipulate your relational database:

- "Customizing Colors" on page 330
- "Customizing States" on page 330
- "Customizing Actions" on page 331

Customizing Colors

By default, planning unit status (for example, Under Review or First Pass) is displayed in black. You can customize each state to display in another color. Valid colors include all named colors that browsers recognize. See the Microsoft Web site for supported named colors.

The information for process status state colors is stored in the `HSP_PM_STATES` table in the `COLOR` column. You must run statements for your relational database to change the color value.

After you customize the colors, restart the Web application server.

Example: SQL Relational Database

Run this query to list all available process status states and their current color settings: `select * from hsp_pm_states`

This returns the `state_id`, the name of the process status state, and the color. By default, the color value is `<NULL>`, which converts to black.

Note the `state_id` of the process status state you want to change, and then run this query:

```
UPDATE HSP_PM_STATES SET COLOR = 'new color' WHERE STATE_ID = state_id
```

Run this SQL statement to change the First Pass process status state color to green:

```
UPDATE HSP_PM_STATES SET COLOR='GREEN' WHERE STATE_ID=1
```

Note: For Oracle relational databases, issue a `COMMIT;` command to commit the transaction.

Customizing States

You can customize these preset process status states:

- Not Started
- First Pass
- Under Review
- Frozen
- Distributed
- Signed Off
- Not Signed Off
- Approved

The process status state information is stored in the `HSP_PM_STATES` table in the `NAME` column. You must run statements for your relational database to change the state value.

Example: SQL Relational Database

Run this query to list all available process status states and their current names:

```
select * from hsp_pm_states
```

This returns the `state_id`, the name of the process status state, and the color.

Note the state_id of the process status state you want to change, and then run this query:

```
UPDATE HSP_PM_STATES SET NAME = 'NewName' WHERE STATE_ID = state_id
```

Run this SQL statement to change the Approved process status state name to Promoted:

```
UPDATE HSP_PM_STATES SET NAME='PROMOTED' WHERE STATE_ID=1
```

Note: For Oracle relational databases, issue a COMMIT; command to commit the transaction.

Customizing Actions

You can customize these preset process status actions:

- Originate
- Start
- Promote
- Submit, Submit to Top
- Exclude
- Reject
- Approve
- Delegate
- Take Ownership
- Freeze, Unfreeze
- Distribute, Distribute Children, Distribute Owner
- Sign Off
- Reopen

The process status action information is stored in the HSP_PM_ACTIONS table in the NAME column. You must run statements for your relational database to change the action value.

Example: SQL Relational Database

Run this query to list all available process status actions and their current names:

```
select * from hsp_pm_actions
```

This returns the action_id and the name of the process status action.

Note the state_id of the process status action you want to change, and then run this query:

```
UPDATE HSP_PM_ACTIONS SET NAME = 'NewName' WHERE ACTION_ID = action_id
```

Run this SQL statement to change the Approve process status action name to Accept:

```
UPDATE HSP_PM_ACTIONS SET NAME='ACCEPT' WHERE ACTION_ID=1
```

Note: For Oracle relational databases, issue a COMMIT; command to commit the transaction.

Creating Spreading Patterns

Administrators who understand SQL 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 85 HSP_SPREAD_PATTERN Table

Column	Description
NAME – VARCHAR (80) UNIQUE NOT NULL	Internal name (not displayed)
LABEL – VARCHAR (80) NOT NULL	Reference to a string ID in the resource file, which displays in the user interface. If a string resource is not created, the LABEL string identifier displays instead (for example “Label_Spread_445”).
PATTERN VARCHAR (255) NOT NULL	Space delimited; one or more elements, up to 255 characters: ● A number: For example, to specify that the parent value splits evenly across its children, specify 1. ● !: Lock; do not change the value of a cell that has “!” as its pattern element. ● P: Parent value being spread. A child receives the exact value the parent receives. ● *: Repeat the element that follows. Use 1* to repeat 1 for every cell, to take up the rest of the space in the applied pattern. If no * exists as part of an element within a pattern, the entire pattern repeats itself instead of a specific element. A pattern cannot contain only the * character, and can have only one element with the * character. For example, 1 2* 3 is allowed, but 1 2* 3* is not.

3 **Save and close the table.**

The new patterns display as spreading options on the Grid Spread and Mass Allocate dialog boxes.

See also the *Oracle Hyperion Planning User's Online Help* and “Examples of Spreading Patterns” on page 332.

Examples of Spreading Patterns

Assume a data form cell having three children cells is affected by a spread. Examples of how patterns are applied to the three child values:

Table 86 Examples of Applied Spreading Patterns

		Cells		
Pattern	Cell 1	Cell 2	Cell 3	**Explanation**

		Cells		

4 4 5	4	4	5	The new value is spread to level zero members using the 4 4 5 pattern. For example, if you set Q1 to 13, its value is spread as:
				Jan = 4
				Feb = 4
				Mar = 5

1	1	1	1	1 is the same as 1*. This FLOW spread type shows how FLOW works if no values exist to make the spread proportional. The parent's value is divided by 3, the number of children cells, and spread equally to each of its three children.
P	P	P	P	Each child cell receives the parent's value. This is more a copy pattern than a spread.
!* P	!	!	P	Equivalent to a BALANCE type of spread.
P !*	P	!	!	Equivalent to a FIRST type of spread.
1 2 1	1	2	1	Similar to a bell-curve type of spread.
0 1 0	0	1	0	The beginning and ending children cells receive no spread values, and the middle child member receives the parent's value.

Creating Instructions for Planners

Using Planning features, administrators can create customized instructions to guide planners in their work:

- Create explicit instructions in the Description text box when creating data forms. For example, “Select this data form to forecast revenue income.” See “Editing Data Forms” on page 135.
- 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.” See “Managing Task Lists” on page 211.
- When setting options for data forms, select the Enable Cell-level Document option (see “Setting Row and Column Layout” on page 125), 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 141.

- 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 74.

14 Troubleshooting

In This Chapter

For detailed troubleshooting information, see the *Oracle Hyperion 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 Online Help* and *Oracle Hyperion 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:

(Number of members * Average length of member names) + Number of members + 500 <= 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.

Data Form Error Message

Scenario:

The first person to use Planning after an Essbase timeout may receive an error that the data 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 73.

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 data 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, Fusion Edition, 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 data 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 Hyperion 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 data 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 data form displays the child, usually above the parent.
2. New data is entered for the child.
3. The data form is saved. The save operation reads the data 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 data forms, you can use these methods to avoid implied shares.

- For a parent and child on the same data form: Add a dummy member as an aggregating child. The dummy member is included in the outline but is not used on data 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 data 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 that, 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 data 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, Fusion Edition, 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:

1 Close Planning.

2 Close Essbase.

3 On the Windows desktop, right-click **My Computer**.

4 Select **Properties**.

5 Select **Advanced**.

6 Select **Environment Variables**.

7 Change the settings for Temp and Tmp in the user variable to the 8.3 naming convention.

For example, `C:\Temp`

8 Open, and open the outline.

9 Close, and restart Planning.

10 Try creating or refreshing the database on the Manage Database page. If the error recurs, repeat the previous steps, but this time also create a folder called `C:\tmp` and set the System Environment variables **Temp** and **Tmp** to `C:\tmp`.

If the error persists, create a `Temp` folder under `C:\temp`. Repeat the previous steps, redirecting first the user variables then the system variables to `C:\temp\temp`. If the error continues, redirect the environment variables to `C:\temp`.

Note: The settings for Temp and Tmp in the user and system variables must be unique. Do not set the user and system variables to the same folder.

Maximum Number of Applications Already Connected or Database is Locked

Scenario:

You are using DB2 as your relational database, and you receive either of these messages when you try to create or refresh the database:

- … maximum number of applications already connected
- Database is locked …

By default, the `MAXAPPLS` parameter is set to 40.

Solution:

Increase the number of applications allowed by increasing the MAXAPPLS parameter.

➤ To increase the `MAXAPPLS` parameter:

1 **In Control Center, right-click the database and select Configure Parameters.**

Alternately, you can set the parameter from the DB2 window.

2 **Set the `MAXAPPLS` parameter, using this format:**

```
db2 -v update db cfg for database name using MAXAPPLS n

db2 -v terminate
```

where *`database name`* is the name of the relational database and *`n`* is the number of applications that can be connected simultaneously. For example, to increase the number of maximum number of applications that can be connected simultaneously to 100 for a database named Business, specify:

```
db2 -v update db cfg for Business using MAXAPPLS 100

db2 -v terminate
```

Optimizing WebLogic Server Parameters

Scenario:

If the following conditions apply to your environment, then check the WebLogic Server settings described in the Solution.

- Your application's performance is slow.
- You have tried the solutions in "Optimizing Performance" on page 81.
- 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 Enterprise Performance Management Workspace, Fusion Edition, select **Navigate**, then **Applications**, then **Planning**. Select a application. If prompted, enter your logon information.

3 Select **Administration**, then **Application**, then **Settings**.

4 From **Show**, select **Advanced Settings**, then click **Go**.

5 Click **Register Shared Services**.

Session Timeout

Scenario:

If a user quits a session by closing the browser instead of logging off, the session times out after a period of time, and the user name is removed from the Statistics page. You can change the number of minutes before the session times out by modifying timeout settings for your Web application server. Modify the `web.xml` file in `HyperionPlanning.ear` or `HyperionPlanning.war`.

Solution:

➤ To change the session timeout setting:

1. **Open the `web.xml` file in `HyperionPlanning.ear` or `HyperionPlanning.war`.**
2. **Modify the timeout setting and save your changes.**

   For example, change the number `60` in the `session-timeout` setting to the number of minutes to use.

   To modify timeout settings, see the documentation for your Web application server.

Slow Performance When Opening Data Forms Using a Dial-Up Connection

Scenario:

Opening a data form using a slow network connection (for example, with a modem) is slow.

Solution:

You can increase the network bandwidth when opening data 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>
    <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 Data Forms

Scenario:

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

Solution:

You can display a warning when users try to open data 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 data form cells that trigger a warning message upon opening:

1 In Planning, select **File**, then **Preferences**.

2 Select **Display Options**.

3 In **Warn If Data 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, a error message might state that the database is not configured correctly. This can result from various causes, such as incorrect configuration for auditing tables.

Unable to Refresh Application Error

Scenario:

When using a DB2 database and refreshing an application in 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 Hyperion 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's Hyperion® Shared Services are running.

Review the error log. (See *Oracle Hyperion Enterprise Performance Management System Installation and Configuration Troubleshooting Guide*.)

If the Log On button does not appear at logon, adjust the security settings in your Web browser, and add the Planning server name as a Trusted Site. For example, in Internet Explorer, select **Tools**, then **Internet Options**, then **Security**, then **Trusted Sites**, and then **Sites** and add `http://`*`servername`* to the trusted sites zone.

A Customizing Data Forms with JavaScript

In This Appendix

About Customizing Data Forms

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

You can customize data forms in these ways:

- Add buttons to data forms that link to Web pages or run custom JavaScript code.
- Modify application behavior when users save data 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 data forms, see the sample code and comments in `SampleValidateData.js`.

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

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 (Planning default logic or customCellEnterPost) runs after this completes.

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

Argument	Description
Row	An integer that defines the row for the cell.
Col	An integer that defines the column for the cell.

Argument	Description
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

`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. Data 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

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

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 data forms are loaded or reloaded. A JavaScript variable called `savedData` indicates whether Save has been completed. For example, when users open data 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 data 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 data forms or run reports.

`SampleValidateData.js` includes an example that adds a button labeled **Validate** to a data 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 data form-level behavior that is launched when users click Save. With default Planning behavior, when users click Save on data 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 tare 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

Return Value	Description
True	Saves the grid.
False	Cancels the save.

Deploying Custom JavaScript

When using custom JavaScript in Planning applications:

- Specify which data 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 data forms are saved, so default calc scripts could overwrite the actions of custom JavaScript. If custom calc scripts run automatically after data 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 data 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 data 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`.

B

Naming Restrictions

In This Appendix

Restrictions for Applications and Databases

When naming applications, 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

    - vertical bars

- For Planning 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 Data 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.
- Distinguish between upper and lower case only if case sensitivity is enabled.
- 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. Essbase ignores such spaces.
- 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 *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 GEN

GENRANGE GROUP GT ID IDERROR INTEGER
LE LEVELRANGE LOCAL LOOPBLOCK
LOOPPARMS LT MBR MBRNAME MBRONLY
MINUS MISSING MUL MULOP NE NON
NONINPUT NOT OR PAREN PARENPARM
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	>	greater than sign
*	asterisk	<	less than sign
@	at sign	()	parentheses
\	backslash	%	percent sign
{ }	brackets	.	period
:	colon	+	plus sign
,	comma	;	semicolon
-	dash, hyphen, or minus	/	slash
!	exclamation point	~	tilde

=	equal sign

In calculation scripts and formulas, enclose member names that are also 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

Planning user names can have up to 80 characters.

C

Data Form Formula Functions

In This Appendix

Using Formulas and Formula Functions

After you create a formula row or column, create the formula in the formula bar that displays above Point of View in the data form Layout page. Formulas include grid references, mathematical operators, and, optionally, mathematical functions (see “Formula Functions” on page 362).

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 Oracle Essbase.

Creating Formulas Using the Formula Bar

The formula bar contains a text box for entering formulas and a drop-down list of functions for building formulas. If you select a grid member, member selection is activated on the formula bar.

➤ To create a formula using the formula bar:

1 **At the top of the page, use the formula bar to create a formula using one of the following methods:**

- Type the formula directly into the formula bar
- Use the drop-down menu on the formula bar to select the formula function.

2 **Click the Validation icon ✓ on the formula bar to validate the formula, and correct any errors.**

Editing Formulas Using the Formula Bar

Use the formula bar to modify a formula within a formula row or column.

➤ To edit a formula using the formula bar:

1. **Open a data form with a formula row or column (see “Selecting and Opening Data Forms and Folders” on page 124).**
2. **Select a formula row or column.**

   The associated formula displays in the formula bar at the top of the page, under the Layout tab.
3. **Modify the formula in the formula bar.**
4. **Click f(-)✓ in the formula bar to validate the formula, and correct any errors.**

Deleting Formulas

Use the formula bar to delete formulas from a formula row or column.

➤ To delete a formula:

1. **Select the appropriate formula row or column, and highlight the formula in the formula bar.**
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 data form formula rows and columns. To insert formula rows and columns in data forms, see “Adding Formula Rows and Columns” on page 132.

The syntax for mathematical functions is:

```
FunctionName(arguments)
```

Table 87 Mathematical Function Syntax

Variable	Description
FunctionName	The name of a mathematical function.
arguments	A numeric value, a row, column, or cell reference, or an embedded function.

Table 88 Mathematical Functions

Function	Description
Abs	Returns the absolute value of numeric values or references.
Average	Returns the average of a group of numeric values or references.
AverageA	Returns the average of a group of numeric values or references. The calculation includes #MISSING cells only for rows or columns that are not suppressed.
Count	Returns the number of values in a group of numeric values or references.
CountA	Returns the number of values in a group of numeric values or references. The calculation includes #MISSING cells only for rows or columns that are not suppressed.
Difference	Returns the absolute value of a numeric value or reference subtracted from another numeric value or reference.
Eval	Evaluates an expression. Eval is useful for embedding expressions as function arguments.
IfThen, If	Returns one value if a condition equals true, and another value if a specified condition equals false.
Max	Returns the maximum value from a group of numeric values or references.
Min	Returns the minimum value from a group of numeric values or references.
Mod	Returns the remainder, modulus, from a division formula.
PercentOfTotal	Returns the result of a numeric value or reference divided by another numeric value or reference, multiplied by 100.
Pi	Returns the number 3.14159265358979, to 15 digits.
Product	Multiplies all numbers or references and returns the product.
Random	Returns a random number between 0.0 and 1.0.
Rank	Returns the highest or lowest value of a specified column or row.
Round	Rounds a number up or down by specified digits.
Sqrt	Returns the square root of a numeric value, row, column, or cell.
Sum	Returns the sum of a group of numeric values or references.
Truncate / Trunc	Removes the specified number of digits from numeric values.
Variance / Var	Evaluates the difference between the specified values based on the account type for the current account.
VariancePercent / VarPer	Evaluates the percentage difference between the specified values based on account type for the current account.

Arguments

Mathematical functions accept numeric values, row, column, or cell references, or embedded functions as arguments. There are four argument types:

- Numeric
- Property
- Row, column, or cell reference
- Embedded Functions

Numeric Arguments

The syntax for a numeric argument is

```
(numeral1, numeral2,...numeraln)
```

where numerals *1* through *n* are any numbers including decimals and negative values. For example, the expression Average(10,20,30) returns the value 20.

Row, Column, or Cell Reference Arguments

The row, column, or cell argument identifies a row, column, or cell in a grid. The syntax is:

```
FunctionName(GridName.GridElement[segment(range)].Property)
```

Table 89 Argument Components

Argument	Description
GridName	The data form name. For example: Difference (grid1.row[5], grid1.row[6]) returns the difference of two rows on data form grid1. Optional. If GridName is not specified, the default is the name of the current data form.
GridElement	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.
segment	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.
range	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[2(3:5)] uses only the 3rd through 5th rows of expanded segment 2. Optional. When *range* is not provided, all expanded cells are used. **Note:** If a segment expands to only one row or column, do not use the range argument.

Argument	Description

property	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])`.

Because segment is the only required part of a reference, the following references are the same:

```
Grid1.row[1].sum
[1]
```

AverageA and CountA include #MISSING cells in the calculation. For example, if row 1 is a segment row that expands to Qtr1 = 100, Qtr2 = 200, Qtr3 = #MISSING, and Qtr4 = 400, the following function returns the value four (4):

```
row[1].CountA
```

All other functions exclude #MISSING data cells. For example, the previous example of row 1 that expands to Qtr1 = 100, Qtr2 = 200, Qtr3 = #MISSING, and Qtr4 = 400, returns three in this example:

```
row[1].Count
```

Property Arguments

Property arguments consolidate expanded references to a single value that is then used in the calculation. Use property arguments to perform calculations on an aggregate row, column, or cell. The two types of property arguments are:

- Aggregate Property Argument (see "Aggregate Property Argument" on page 365)
- Reference Property Argument (see "Reference Property Argument" on page 366)

Aggregate Property Argument

An aggregate row, column, or cell contains multiple rows, columns, or cells, respectively.

The aggregate property argument is the last argument in the following mathematical function syntax:

```
FunctionName(GridName.Gridelement[segment(range)].property)
```

You apply the following aggregate properties to a row, column, or cell reference.

Table 90 Aggregate Properties

Properties	Description
Average	Returns the average of a row, column, or cell. The calculation excludes #MISSING and #ERROR values.
AverageA	Returns the average of a row, column, or cell. The calculation includes #MISSING and #ERROR values.
Count	Returns the number of values in a row, column, or cell. The calculation excludes #MISSING and #ERROR values.
CountA	Returns the number of values in a row, column, or cell. The calculation treats #MISSING and #ERROR values as zero (0).
Max	Returns the maximum value of a row, column, or cell.
Min	Returns the minimum value of a row, column, or cell.
Product	Returns the product of rows or columns.
Sum	Returns the sum of a row, column, or cell.

When used as a mathematical function argument, the default for property is the same as the function. In the following example the default property is Average:

```
Average(row[2])
```

When not used as a mathematical function argument, the default for property is sum. In the following example the default property is the sum of an aggregate row:

```
row[2]
```

Reference Property Argument

A reference property argument specifies how to treat formula reference results and is used in conjunction with the other properties.

There is one reference property argument: `IfNonNumber/IFFN`.

IfNonNumber specifies a replacement of #MISSING and #ERROR values with a specific numeric value. The syntax is:

`AXIS[segment(range)].IfNonNumber(`*`arg`*`).AggregateProperty`

Argument	Description
AXIS	One of these keywords: row, column, or cell. Optional.
Segment(range)	Indicates any valid axis reference, such as a row number, column letter.
IfNonNumber	Indicates how to treat missing or error data within the Axis Ref.
(arg)	Indicates what number to use if missing or error data is encountered within the AxisRef.
AggregateProperty	The aggregate function is used for aggregate segments. Optional.

For example:

If cell[1,A] = 3 and

cell[1,B] = #MISSING,

The expression:

```
cell[1,A] / cell[1,B]
```

returns #ERROR.

The expression:

```
cell[1,A] / cell[1,B].ifNonnumber(1)
```

replaces cell[1,B] with 1 and returns a 3.

Note: If you use suppression for #MISSING in a grid, and the grid contains a formula row or column that uses the IfNonNumber property, #MISSING remains suppressed.

Embedded Functions as Arguments

You can embed functions as arguments within a function.

Example:

In this example, the function Average is embedded in the function Sum:

sum(row[3:5], avg(row[4:6], 40, 50), row[7; 9], 70, 80)

- Row segments 3, 4 and 5
- The average of row segments 4, 5 and 6, with the numbers 40 and 50
- Row segments 7 and 9
- The numbers 70 and 80

Abs

Abs is a mathematical function that returns the absolute value of a numeric value, row, column, or cell. The absolute value of a number is that number without regard to sign. A negative number becomes positive, while a positive number does not change. The function syntax is:

```
Abs (argument)
```

where *argument* is one of the following:

Argument	Description
numeric	A numeric value. For example, Abs (- 20) returns the value 20. Numeric values can include decimals and negative values.
row, column, or cell reference	A pointer to a row, column, or cell within a grid. References can be specified in several ways. The reference syntax is: `GridName.GridElement[segment(range)].Property.`

Argument	Description
function	An embedded function.

Examples:

The following expression includes a numeric argument and returns the value 30:

```
Abs ( -30 )
```

The following example returns the absolute value of the value in row 1:

```
Abs(row[1])
```

The following example calculates the absolute value of the sum of column E:

```
Abs( column[E].sum )
```

The following example points to expanded rows 1 through 3 within design segment 3 of the data 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:

```
Average(arguments) or Avg(arguments)
```

where *arguments* is one or more of the following:

Argument	Description
numeric	A numeric value. For example, Average (10, 20, 30) returns the value 20. Numeric values can include decimals and negative values.
row, column, or cell reference	A pointer to a row, column, or cell within a grid. References can be specified in several ways. The reference syntax is: `GridName.GridElement[segment(range)].Property` For example Avg(Grid1.row[4(3:5)]) returns the average of data form grid1, row 4, range 3 through 5.
function	An embedded function.

Examples:

The following expression returns the value 20:

```
Avg( 10, 30, 20)
```

The following example returns the average of all numbers that are part of three aggregate rows:

```
Average( row[1], row[6], row[8] )
```

The following example calculates the average of three aggregate columns; E, G, and I. The calculation produces three numbers, then calculates the average of the three numbers:

```
Avg(column[E].avg, column[G].avg,column[I].avg)
```

The following example calculates the average of aggregate row 3 and divides the average by 100:

```
Avg(row[3])/100
```

AverageA

AverageA is a mathematical function that returns the average of a group of numeric values, rows, columns, or cells. AverageA includes #MISSING and #ERROR cells, which are treated as zero values when obtaining the average.

Note: #MISSING and #ERROR are included only for rows or columns that are not suppressed.

The function syntax is:

```
AverageA(arguments) or AvgA(arguments)
```

where *arguments* is one or more of these arguments:

Argument	Description
numeric	A numeric value. For example, AverageA (10, 20, 30) returns the value 20. Numeric values can include decimals and negative values.
row, column, or cell reference	A pointer to a row, column, or cell. References can be specified in several ways. The reference syntax is: `GridName.GridElement[segment(range)]. Property.` For example, AvgA(Grid1.row[4(3:5)] returns the average of data form grid1, row segment 4, range 3 through 5.
function	An embedded function.

Example

In the following example, if a grid has 4 rows with the values 10, 20, 30, and, #ERROR. The following formula in the 5th row returns the value 15:

```
AverageA([1:4])
```

Count

Count is a mathematical function that returns the number of values in a group of numeric values, rows, columns, or cells. Count excludes #MISSING and #ERROR when obtaining the count. The function syntax is:

```
Count (arguments)
```

where *arguments* is one or more of these arguments:

Argument	Description
numeric	A numeric value. For example, Count (10, 20, 30) returns the value 3. Numeric values can include decimals and negative values.
row, column, or cell reference	A pointer to a row, column, or cell within a grid. References can be specified in several ways. The reference syntax is: `GridName.GridElement[segment(range)].Property`
function	An embedded function.

Examples:

The following example returns the count of three rows, 1, 6, and 8:

```
Count(row[1], row[6], row[8])
```

The following example returns the count of 3 columns:

```
Count( column[E], column[G], column[I] )
```

The following example calculates the count of the cell located at row 4, column D:

```
Count(cell[D,4])
```

The following example calculates the count of aggregate row 3 in grid 5:

```
Count(grid1.row[3])
```

CountA

CountA is a mathematical function that returns the number of values in a group of numeric values, rows, columns, or cells. CountA includes #MISSING and #ERROR cells when obtaining the count only for rows or columns that are not suppressed. The function syntax is:

```
CountA(arguments)
```

where *arguments* is one or more of these arguments:

Argument	Description
numeric	A numeric value. For example, `CountA(10,20,30,50)` returns the value 4. Numeric values can include decimals and negative values.
row, column, or cell reference	A pointer to a row, column, or cell within a grid. References can be specified in several ways. The reference syntax is: `CountA(GridName.GridElement[segment(range)]).property`
function	An embedded function.

Example

In the following example, if a grid has 4 rows with the values 10, 20, 30, and, #ERROR. The following formula in the 5th row returns the count of four rows:

```
CountA([1:4])
```

The following example returns the count of four rows:

```
CountA(row[1], row[6], row[8] row[where data yields #ERROR])
```

Difference

Difference is a mathematical function that returns the absolute value of the difference of a numeric value, row, or column subtracted from another numeric value, row, or column. This is also known as the variance. The function syntax is:

Difference(arg1, arg2)

where *arg2* is subtracted from *arg1* and are one or more of the following arguments:

Argument	Description
numeric	A numeric value. For example, Difference (3, 5) returns the absolute value 2. Numeric values can include decimals and negative values.
row, column, or reference	A pointer to a row, column, or cell within a grid. References can be specified in several ways. The reference syntax is: `GridName.GridElement[segment(range)].Property` The following example returns the difference of two rows in data form grid1: Difference(grid1.row[1], grid1.row[6])
function	An embedded function.

Note: The Difference function returns the absolute value of arg2 subtracted from arg1, whereas the minus sign in subtraction negates a number.

Examples:

The following example returns the absolute value of 8:

```
Difference(3, -5)
```

The following example calculates the difference of two aggregate columns:

```
Difference( column[E], column[G] )
```

Note: You can type the text label “Difference” or “Variance.”

Eval

Eval is a mathematical function that evaluates an expression. You use Eval as an embedded function argument to consolidate multiple expressions into one expression. The function syntax is:

Eval(*expression*)

where *expression* is one or more of these arguments:

Argument	Description
numeric	A numeric value. Numeric values can include decimals and negative values.
row, column, or reference	A pointer to a row, column, or cell within a grid. References can be specified in several ways. The reference syntax is: `GridName.GridElement[segment(range)].Property`
function	An embedded function.
operators	Use any of the supported arithmetic operators (+, -, *, /, ^, %, and so on).

Example

The following example divides row 1 by row 2 and then rounds the data to 4 places:

```
Round(Eval([1]/[2]),4)
```

IfThen, If

IfThen is a conditional function that returns a value when the condition equals True, and another value when the condition equals False.

The function syntax is as follows:

```
IfThen(Condition, TrueParameter, FalseParameter)
```

- *Condition* is a logical expression that evaluates to true or false. Full conditional logic can be used as well as complex Boolean operators (And, Not, and Or). A *condition* can also test for #MISSING and #ERROR values. See the following table for a list of valid conditional operators.
- *TrueParameter* and *FalseParameter* are any valid expression that are evaluated based on the outcome of the condition.

The following table describes the conditional operators that are fully supported. Alternate syntax is listed wherever it is supported by a conditional operator.

Table 91 Conditional Operators

Conditional Operator	Syntax	Logic
Equal To	*expression = expression*	Tests if the left expression is equal to the right expression. **Note:** The routine that evaluates the condition does not consider any rounding. If rounding is required, use the Round function. Example: `1= 4` Returns false

Conditional Operator	Syntax	Logic
Greater Than	*expression* > *expression*	Tests if the left expression is greater than the right expression. **Example:** `1 > 4` Returns false
Greater Than or Equal To	*expression* >= *expression*	Tests if the left expression is greater than or equal to the right expression. **Note:** The correct syntax is “>=”. The syntax “=>” is not supported. **Example:** `1 >= 4` Returns false
Less Than	*expression* < *expression*	Tests if the left expression is less than the right expression. **Example:** `1 < 4` Returns true
Less Than or Equal To	*expression* <= *expression*	Tests if the left expression is less than or equal to the right expression. **Note:** The correct syntax is “<=”. The syntax “=<” is not supported. **Example:** `1 <= 4` Returns true

Not Equal To	*expression* <> *expression*
	expression != *expression*
		Tests if the left expression is not equal to the right expression.
		Note: The routine that evaluates the condition does not consider any rounding. If rounding is required, use the Round function.
		Example:
		`1 <> 4`
		Returns true
		`1 != 4`
		Returns true

IsMissing	IsMissing (*reference*) IsMiss (*reference*)	Tests if the reference contains a #MISSING result. **Note:** If the reference is an expanded row or column, then all resulting cells must be #MISSING in order for the condition to be true. **Example**: `IsMissing([1])` Returns true if row 1 has a #MISSING value.

Conditional Operator	Syntax	Logic
IsError	IsError (*reference*) IsErr (*reference*)	Tests if the reference contains an #ERROR result. **Note:** If the reference is an expanded row or column, all resulting cells must be #ERROR in order for the condition to be true. Only formula rows and columns can result in #ERROR. Example: `IsError([2])` Returns true if row 2 has a #ERROR value.
IsNonNumeric	IsNN (*reference*) IsNonNumerid (*reference*) IfNN (*reference*) IfNonNumber (*reference*)	Tests if the reference contains a #MISSING or #ERROR results. **Note:** If the reference is an expanded row or column, all resulting cells must be #MISSING and/or #ERROR in order for the condition to be true. **Example:** `IsNN([3])` Returns true if row 3 has a #MISSING or #ERROR value.
Parenthesis	(*condition*)	Used to group a condition. Mostly used for visual clarity. **Example:** `(1 > 4)` Returns false

Table 92 Conditional Operators

Complex Conditions	Syntax	Logic
And	(*condition* AND *condition*) (*condition & 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*) ! (*condition*)	Used to negate the result by reversing the result of the condition. **Example:** `Not (1 > 4)` Returns true
Or	(*condition* OR *condition*) (*condition* | *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:

Argument	Description
numeric	A numeric value. For example, Max (10, 20, 30) returns the value 30. Numeric values can include decimals and negative values.
row, column, or cell reference	A pointer to a row, column, or cell within a grid. References can be specified in several ways. The reference syntax is: `GridName.GridElement[segment(range)].Property.`
function	An embedded function.

Examples:

The following example returns the maximum value in rows 1, 6, and 8:

```
Max(row[1], row[6], row[8])
```

The following example calculates the maximum of the sums of aggregate rows:

```
Max(row[1].sum, row[2].sum, row[3].sum)
```

Min

Min is a mathematical function that returns the minimum value in a group of numeric values, rows, columns, or cells. The function syntax is as follows:

```
Min (arguments)
```

where *arguments* is one or more of these arguments:

Argument	Description
numeric	A numeric value. For example, Min (10, 20, 30) returns the value 10. Numeric values can include decimals and negative values.
row, column, or cell reference	A pointer to a row, column, or cell within a grid. References can be specified in several ways. The reference syntax is: `GridName.GridElement[segment(range)].Property`
function	An embedded function.

Examples:

The following example returns the minimum value in rows 1, 6, and 8:

```
Min (row[1], row[6], row[8])
```

The following example calculates the minimum of the sums of aggregate rows:

```
Min(row[1].sum, row[2].sum, row[3].sum)
```

Mod

Mod is a mathematical function that returns the remainder, or modulus, from a division. The function syntax is:

Mod (*arg1, arg2*)

where *arg2* is the divisor and *arg1* and *arg2* are one of these arguments:

Argument	Description
numeric	A numeric value. For example, Mod (6, 4) returns the value 2. Numeric values can include decimals and negative values.
row, column, or cell reference	A pointer to a row, column, or cell within a grid. References can be specified in several ways. The reference syntax is: `GridName.GridElement[segment(range)].Property`
function	An embedded function.

Example:

The following example divides 10 by 5 and returns the remainder of 0:

```
Mod (10,5) = 0
```

PercentOfTotal

PercentOfTotal is a mathematical function that returns the result of a numeric value, row, column, or cell divided by another numeric value, row, column, or cell which is multiplied by 100. The function syntax is:

```
PercentOfTotal (arg1, arg2)
```

- where *arg1* is a component of the running total (*arg2*). Normally, this is a row or column reference.
- where *arg2* is the running total relative to *arg1*. Normally this is a cell reference containing the grand total.
- *arg1* is divided by *arg2*, with the result multiplied by 100. *Arg1* and *arg2* are one or more of these arguments:

Argument	Description
numeric	A numeric value. For example, PercentofTotal (100, 20) returns the value 500. Numeric values can include decimals and negative values.
row, column, or cell reference	A pointer to a row, column, or cell within a grid. References can be specified in several ways. The reference syntax is: `GridName.GridElement[segment(range)].Property`
function	An embedded function.

Note: This function requires two arguments.

Examples:

The following example returns the value of 5 percent.

```
PercentofTotal(20, 400)
```

The following example divides the value of each cell in column A by the Total Mkt value in cell A5, multiplies the result by 100, and displays the resulting PercentOfTotal in column B. The formula:

```
PercentOfTotal ([A], [A,5]),
```

Using the above example, the following table shows the PercentOfTotal results in column B:

		A	B
1		Sales	% Total
2	Mkt1	60	**20%**
3	Mkt2	120	**40%**

		A	B
4	Mkt3	120	**40%**
5	Total Mkt	300	**100%**

Tip: You enter the formula by clicking on the header for Column B and using the formula bar.

Pi

Pi is a mathematical function that returns the number 3.14159265358979, the mathematical constant, accurate to 15 digits. Pi is the ratio of the circumference of a circle to its diameter. The function syntax is:

`PI()`

Example:

The following example returns row 3 divided by the product of Pi and 2:

```
row[3] / (PI() * 2)
```

Product

Product is a mathematical function that multiplies all numbers or references and returns the product. The function syntax is:

Product(*`arguments`*)

where *arguments* is one or more of these arguments:

Argument	Description
numeric	A numeric value. For example, Product(2, 20) returns the value 40. Numeric values can include decimals and negative values.
row, column, or cell reference	A pointer to a row, column, or cell within a grid. References can be specified in several ways. The reference syntax 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, Fusion Edition and does not depend on the database connection. The function syntax is as follows:

Rank([Reference], Order)

Rank([Reference], Order, Unique)

Argument	Description
Reference	The range of cells, rows, or columns to rank, with letters identifying columns and numbers identifying rows. For example, specify [A,1:5] to rank the values for rows 1 through 5 in column A. You can use the .ifNN property with a range of cells to assign numbers to any cells with nonnumeric values so that those cells can be ranked. For example, you can use .ifNN(-1) to assign the value -1 to any cell with a missing value.
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:

	East	Rank
Cola	16	2
Fruit Drinks	23	1
Beer	16	2
Diet	missing	missing
Root Beer	0	4

When two values are equal, they receive the same rank value. In the example above, Cola and Beer have the same value and therefore the same rank.

This formula in column B assigns the value of -1 to any nonnumeric value so it can be ranked:

```
Rank([A,1:5].ifNN(-1), descending)
```

In the following result, the missing value now has a rank of 5:

	East	Rank
Cola	16	2
Fruit Drinks	23	1
Beer	16	2
Diet	missing	5
Root Beer	0	4

Example:

The following example builds on the previous example explaining how the unique parameter affects the results:

This formula in column B assigns the value of -1 to any nonnumeric value so it can be ranked, and also indicates that each ranking should be unique:

```
Rank([A,1:5].ifNN(-1), descending, true)
```

In the following result, the missing value now has a rank of 5, and Beer has a value of 3 (even though it has the same data value as Cola):

	East	Rank
Cola	16	2
Fruit Drinks	23	1
Beer	16	3
Diet	missing	5

	East	Rank
Root Beer	0	4

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:

Argument	Description
numeric	A numeric value. For example, Round(81.3987,3) returns the value 81.399. Numeric values can include decimals and negative values.
row, column, or cell reference	A pointer to a row, column, or cell within a grid. References can be specified in several ways. The reference syntax is: `GridName.GridElement[segment(range)].property`
function	An embedded function.

Integer specifies the number of digits to which you want to round the number:

- If *integer* is greater than zero, the number rounds to the specified number of decimal places.
- If *integer* is zero, the number rounds to the nearest integer.
- If *integer* is less than zero, the number is rounded to the left of the decimal point.

Examples:

The following example rounds to 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:

Argument	Description
numeric	A numeric value. For example, Sqrt(100) returns the value 10. Numeric values can include decimals and negative values.
row, column, or cell reference	A pointer to a row, column, or cell within a grid. References can be specified in several ways. The reference syntax is: `GridName.GridElement[segment(range)].property`
function	An embedded function.

Example

The following example returns the value of 4:

```
Sqrt(16)
```

Sum

Sum is a mathematical function that returns the summation of a group of numeric values, rows, columns, or cells.

The syntax for the Sum function is:

Sum (arguments)

where *arguments* is one or more of these arguments:

Argument	Description
numeric	A numeric value. For example, Sum(10, 20, 30) returns the value 60. Numeric values can include decimals and negative values.
row, column, or cell reference	A pointer to a row, column, or cell within a grid. References can be specified in several ways. The reference syntax is: `GridName.GridElement[segment(range)].property`
function	An embedded function.

Examples:

The following example returns the value 30:

```
sum(10,20)
```

The following example returns the sum of three rows:

```
sum(row[1],row[6],row[8])
```

The following example calculates the sum of three aggregate columns:

```
sum(column[E], column[G], column[I])
```

Truncate / Trunc

Truncate is a mathematical function that removes the specified number of digits from numeric values.

Syntax:

```
Trunc (arg1, integer)
```

- where `arg1` is one of these arguments:

Argument	Description
numeric	A numeric value. For example, 234.567.
row, column, or cell reference	A pointer to a row, column, or cell within a grid. References can be specified in several ways. The reference syntax 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 93 Expected Results when Using the Variance Function

Column A	Column B	Var ([A] , [B])=0	Var ([A] , [B])>0	Var ([A] , [B])<0
Asset	Asset	0	Returns a positive value	Returns a negative value
Liability	Liability	0	Returns a positive value	Returns a negative value
Equity	Equity	0	Returns a positive value	Returns a negative value
Revenue	Revenue	0	Returns a positive value	Returns a negative value
Expense	Expense	0	Returns a negative value	Returns a positive value

Variance Behavior

- The Variance function expects comparison of the same account type. When you compare two different account types, like Sales & Expense, the Variance function performs the straight math without applying the logic of the account type. For example:

Sales	Expense	Result
-400	100	-500

- When the Variance function is applied to a dimension that is not tagged as an Accounts type, an #ERROR results at runtime.
- #MISSING is treated as zero (0), unless specified differently using the ifnonnumber property.

Examples

The Variance function accepts cell, column, or row references only.

Syntax	Example
Sample syntax referencing a column:	`Var ([A], [B])`
Sample syntax referencing a row:	`Var ([3], [4])`

Syntax	Example
Sample syntax referencing a cell:	`Var (Cell [3,A], [3,B])`

In this example, the variance between column A (**Actual**) and column B (**Budget**) is calculated as:

```
Var([A],[B])
```

This example produces the following report:

```
                               Year   Product Market
                      Actual   Budget Variance
                    ======== ======== ========
Sales(Income)       400,855  373,080    27,775
COGS (Expense)      179,336  158,940   -20,396
```

VariancePercent / VarPer

VariancePercent is a financial function that evaluates the difference, in percent, between the specified values based on account type for the current account. For example, for an Income, Flow, Asset, or Balance account, a positive result represents an increase, so the result appears as a positive number. For Expense or Liability accounts, a positive result represents a decrease, so the result appears as a negative number.

Syntax:

```
VarPer (reference1, reference2)
```

where *reference1* and *reference2* are references to a row, column, or cell that correspond to members of the same Account dimension whose VariancePercent results are to be calculated.

Expected Results

The following table describes the expected results when using the VariancePercent function with Accounts tagged with the below UDAs.

Table 94 Expected Results when Using the VariancePercent Function

Col A	Col B	VarPer ([A] , [B])=0	VarPer ([A] , [B])>0	VaPer ([A] , [B])<0
Asset	Asset	0	Returns a positive value	Returns a negative value
Liability	Liability	0	Returns a negative value	Returns a positive value
Equity	Equity	0	Returns a positive value	Returns a negative value
Revenue	Revenue	0	Returns a positive value	Returns a negative value
Expense	Expense	0	Returns a negative value	Returns a positive value

VariancePercent Behavior

- The VariancePercent function expects comparison of the same account type. When you compare two different account types, like Sales & Expense, the VariancePercent function performs the straight math without applying the logic of the account type. For example:

Sales	Expense	Result
-400	100	-5.

- When the VariancePercent function is applied to a dimension that is not of type Accounts, an #ERROR results at runtime.
- #MISSING is treated as zero (0), unless specified differently using ifnonnumber property.

Examples:

The VariancePercent function accepts, cell, column, or row references only.

Syntax	Example
Sample syntax referencing a column:	`VarPer ([A], [B])`
Sample syntax referencing a row:	`VarPer ([3], [4])`
Sample syntax referencing a cell:	`VarPer (Cell [3,A], [3,B])`

In this example, the VariancePercent between column A (**Actual**) and column B (**Budget**) is calculated as follows:

```
VarPer([A],[B])
```

This example produces the following report:

```
                       Year Product Market
                      Actual   Budget VariancePercent
                    ======== ======== ========
Sales(Income)       400,855  373,080     7%
COGS (Expense)      179,336  158,940   -13%
```

D

Accessibility Features

In This Appendix

Enabling Accessibility

Accessibility features in Planning are on by default, and are always available.

Accessibility Features

Planning provides these accessibility features. For information about the recommended software configuration, see the *Oracle Hyperion Enterprise Performance Management System Certification Matrix*. This matrix is posted on the Oracle Fusion Middleware Supported System Configurations page on OTN (listed in the Oracle Business Intelligence product area): http://www.oracle.com/technology/software/products/ias/files/fusion_certification.html

- Support for screen readers and magnifiers
- Keyboard navigation equivalents for all objects
- Alt+Z to skip navigation links and read the Planning main content area. Note that Ctrl+F6 does not read this area
- Alt and Title attributes for all objects, images, and page titles
- Support for expanding and collapsing trees using the Enter key and keyboard equivalents
- Data form access using keyboard equivalents
- Tab indexing, implemented from left to right on all pages

Note: If you are using JAWS® Screen Reading Software, we recommend using the Internet Explorer browser.

Using Keyboard Equivalents

These sections list the keyboard equivalents implemented for menus and tasks completed on pages and dialog boxes.

- "General Navigation" on page 388 contains the keyboard equivalents used throughout Planning.
- "Tasks" on page 396 presents the keyboard equivalents that are specific to each task, including exceptions to the default keyboard equivalents for buttons such as Add and Delete.

Note: Some options and movements within Planning do not need an assigned hot key. In these cases, use the Tab key for navigation.

Note: If you use a version of the Mozilla Firefox browser later than release 1.5, substitute Alt +Shift for Alt in the keyboard equivalents.

General Navigation

Oracle Hyperion Planning, Fusion Edition screens use common keyboard equivalents for screen navigation. For example, pressing the up arrow key moves the cursor up through a list, and pressing Tab moves focus from one screen element to another. Also, keyboard equivalents for buttons such as Cancel and Save are the same in most screens.

Note that:

- You can also press Tab or Shift+Tab to move between page or dialog box elements. If focus is on the last element in a page or dialog box, press Alt+Z to move focus to the first element.
- Some EPM Workspace global keyboard equivalents such as Ctrl+G do not work when focus is within a data form cell. When this occurs, press Ctrl+Shift+F6, then Shift+Tab, and then the navigation keys.
- If you edit data cell contents, then copy or cut the cell contents, the focus moves away from the cell. To return the focus to the cell using the keyboard, press Alt+Z, and then tab to the cell.
- Focus cannot be moved to the previous cell of a grid when focus is in the last cell of the grid. When this occurs, press Shift+ End to move focus to the last cell of the first row of the grid, then press Ctrl+End to move focus to the last cell of the grid, and then use the Shift+Tab or arrow keys to move back a cell.

  Exception: Focus cannot be moved to the last cell of the first row of the grid or the last cell of the entire grid if the Shift+End and Ctrl+End keys are used in a cell with data type **Text**. To use these keys, move focus away from the text cell.

The following tables list the general navigation keys, and present the common keyboard equivalents for buttons used throughout the software. Exceptions to the common button keyboard equivalents are listed in the task-related tables (see "Tasks" on page 396).

To navigate within the Planning screen, use these keyboard equivalents.

Focus	Keyboard Equivalent
Main Content Area	Alt+Z
Available List	Alt+U
Selected List	Alt+V
Move up in a list	Up arrow key
Move down in a list	Down arrow key
Move left in a list	Left arrow key
Move right in a list	Right arrow key
Open a grid drop-down menu	Alt+'
Select a grid drop-down menu item	Use up and down arrow keys to move through the list, and then press Enter to select an option
Move to data forms list	Ctrl+0 (zero)
Expand a list of data forms or data form folders	Right arrow key
Collapse a list of data forms or data form folders	Left arrow key
Display the list of data forms in a data form folder	Select the data form folder, and then press Enter
Select a data form from a data form folder list	Use the up and down arrow keys to locate the data form, and then press Enter
Adjust row height and column width in a data form	Select a data form, then navigate to the **Layout** tab, then Tab to **Default column width** or **Default row height** (under **Grid Properties**), then use the arrow keys to select the new defaults, and then press Enter
Move focus from the right content pane to the left content pane	Ctrl+Shift+F6, Shift+Tab, and then Ctrl+0 (zero)
Move focus to the left pane when focus is on a cell associated with a Smart List	Ctrl+1, Ctrl+Shift+F6, Shift+Tab, and then Ctrl+0 (zero)
Move to first cell in grid	Ctrl+Home
Move to last cell of first row of grid	Shift+End
Move to last cell of grid	Ctrl+End
Open a shortcut menu	Shift+F10
Open a business rule	Navigate to the business rule, and then press Enter
View error details in a business rule launch window	Alt+8
Move focus to a task list, basic mode	Ctrl+Shift+F6, then Shift+Tab, and then Ctrl+0 (zero)
Select a task from a task list	Use the up and down arrow keys to locate the task, and then press Enter

Focus	Keyboard Equivalent
Move focus to Adjust, Grid Spread, or Mass Allocate data entry dialog	Alt+2
Move a member up in a dimension hierarchy	Ctrl+1
Move a member down in a dimension hierarchy	Ctrl+2
Expand All	Alt+E
Collapse All	Alt+C
Move to open file upload dialog	Alt+Tab
Read the calculation for read-only cells in data-entry pages	Insert+Numpad 5
Move focus to the left pane when focus is on a cell associated with a Smart List	Ctrl+1, Ctrl+Shift+F6, Shift+Tab, and then Ctrl+0 (zero)

To activate Planning buttons, such as Add and Delete, use these keyboard equivalents.

Button	Keyboard Equivalent
Add	Alt+2
Add All	Alt+5
Add Child	Alt+C
Add Sibling	Alt+L
Assign Access	Alt+G
Cancel	Alt+L
Clear	Alt+C
Close	Alt+Y
Copy	Alt+P
Create	Alt+C
Create Report	Alt+C
Cut	Alt+X
Delete	Alt+9
Edit	Alt+I
End	Alt+7
Finish	Alt+7
Help	Alt+H

Button	Keyboard Equivalent
Go	Alt+G
Move	Alt+M
Move Up	Alt+M
Move Down	Alt+W
New	Alt+W
Next	Alt+6
OK	Alt+O
Paste (within the main window)	Alt+7
Paste (within a shortcut menu)	Alt+V
Preview	Alt+W
Previous	Alt+P
Refresh	Alt+R
Remove	Alt+4
Remove All	Alt+6
Rename	Alt+B
Reset	Alt+R
Restore	Alt+R
Save	Alt+S
Save As	Alt+A
Save Selections	Alt+S
Start	Alt+S
Submit	Alt+S

Main Menu

Use these keyboard equivalents to open the menus available from the menu bar.

Menu	Keyboard Equivalent
File Menu	Alt+F
Edit Menu	Alt+E

Menu	Keyboard Equivalent
View Menu	Alt+V
Administration Menu	Alt+A
Tools Menu	Alt+T
Help Menu	Alt+H

File Menu

These keyboard equivalents are used in the File menu.

File Menu Item	Keyboard Equivalent
New, Data Form	N, F
New, Report	N, R
New, Task List	N, T
New, User Variable	N, U
New, Smart List	N, L
New, Menu	N, M
Open	O
Open, URL	O, U
Open, Application	O, P
Open in Smart View	V
Close	C
Close, Current	C, C
Close, All	C, A
Close, Others	C, O
Preferences	F
Logout	G
Exit	X

Edit Menu

These keyboard equivalents are used in the Edit menu.

Edit Menu Item	Equivalent
Adjust	J
Grid Spread	G
Mass Allocate	L
Comment	O
Lock/Unlock Cells	K
Supporting Detail	N
Add/Edit Document	E
Open Document	C
Annotate Planning Unit	T
Drill Through	D

View Menu

These keyboard equivalents are used in the View menu in Basic mode.

View Menu Item	Equivalent
Basic Mode	B
Task List, Task List	T, T
Task List, Report	T, R
Task List, Status	T, S
Edit Account Annotations	A
Instructions	I
Currency	C
Refresh	R
View Masthead	Ctrl+Alt+0 (zero)
View Pane	Ctrl+Alt+1

Tools Menu

These keyboard equivalents are used in the Tools menu.

Tools Menu Item	Equivalent
Manage Approvals	L
Copy Version	V
Business Rules	R
Job Console	S
Copy Link	K
Custom Links	O
Export as Spreadsheet	E
Ad Hoc, New Ad Hoc Grid	A, C
Ad Hoc, Analyze	A, A
Ad Hoc, Save Ad Hoc Grid	A, S
Ad Hoc, Ad Hoc Options	A, O
Ad Hoc, Zoom In, Zoom in Next Level	A, Z, N
Ad Hoc, Zoom In, Zoom in All Levels	A, Z, V
Ad Hoc, Zoom In, Zoom in Bottom Level	A, Z, B
Ad Hoc, Zoom Out	A, M
Ad Hoc, Remove Only	A, R
Ad Hoc, Keep Only	A, K

Administration Menu

These keyboard equivalents are used in the Administration menu.

Administration Menu Item	Equivalent
Manage, Data Forms and Ad Hoc Grids	M, F
Manage, Menus	M, M
Manage, Smart Lists	M, L
Manage, Task Lists	M, T
Manage, User Variables	M, U
Manage, Business Rule Security	M, B
Manage, Clear Cell Details	M, C

Administration Menu Item	Equivalent
Manage, Copy Data	M, Y
Manage, Dimensions	M, D
Manage, Alias Tables	M, A
Manage, Currency Conversions	M, V
Manage, Exchange Rates	M, X
Manage, Security Filters	M, S
Application, Broadcast Messages	A, M
Application, Properties	A, P
Application, Reports	A, R
Application, Statistics	A, T
Application, Settings	A, S
Application, Create	A, E
Application, Delete	A, D
Application, Register	A, G
Application, Create Database	A, C
Application, Refresh Database	A, B
Approvals, Planning Unit Hierarchy	P, P
Approvals, Scenario and Version Assignment	P, A
Approvals, File Based Import/Export	P, I
Map Reporting Application	R
Data Source, Create	D, C
Data Source, Edit	D, E
Data Source, Delete	D, D
Data Load Settings	L
Initialization, Workforce	I, W
Initialization, Capital Asset	I, C

Help Menu

These keyboard equivalents are used in the Help menu.

Help Menu Item	Equivalent
Help on this Topic	T
Contents	C
Contents Capital Asset Planning	L
Contents Workforce Planning	W
Contents Public Sector Planning and Budgeting	P
Technical Support	S
EPM Documentation	E
About Oracle Enterprise Performance Management System Workspace, Fusion Edition	A

Tasks

Use these keyboard equivalents in the main content area, on pages, and in dialog boxes. See "General Navigation" on page 388 for navigation keyboard equivalents, such as moving up and down within lists, and keyboard equivalents for common buttons, such as Cancel and Save.

Table 95 Keyboard Equivalents for Edit Menu Tasks

Edit Menu Tasks	Keyboard Equivalent
Adjust Data	Submenu Items/Buttons
Adjust Data Button	Alt+4
Close Adjust Data dialog	Esc
Grid Spread	Submenu Items/Buttons
Adjust Data	Alt+7
Adjust Button	Alt+8
Spread Button	Alt+1
Supporting Detail	Submenu Items/Buttons
Add Sibling	Alt+1
Delete All	Alt+A
Promote	Alt+P
Demote	Alt+E

Edit Menu Tasks	Keyboard Equivalent
Duplicate Row	Alt+R
Yes, Set Value to Missing	Alt+Y
No, Leave Value As Is	Alt+N

Table 96 Keyboard Equivalents for View Menu Task List Tasks

Viewing Task Lists Tasks	Keyboard Equivalent
Task List, Task List	T, T
Task List page	Submenu Items/Buttons
Available Task Lists	Alt+1
Task List (in Task List page)	Alt+2
Status	Alt+3
Task List (in Task List page)	Submenu Items/Buttons
Completed, Incomplete	Alt+T
Task List, Report	Submenu Items/Buttons
Add	Alt+2
Add All	Alt+4
Remove	Alt+3
Remove All	Alt+5
Available List Box Selection	Alt+6
Selected List Box Selection	Alt+7
Next	Alt+W
Previous	Alt+M
Create Report	Alt+W
Task List, Status	Submenu Items/Buttons
Next Incomplete	Alt+6
View Task List	Alt+7
View Task List	Submenu Items/Buttons
Display All Completed Dates	Alt+W
Status Link	Alt+M

Viewing Task Lists Tasks	Keyboard Equivalent
Next Incomplete	Submenu Items/Buttons
Complete	Alt+1
Previous Incomplete	Alt+2
Previous	Alt+3
Next	Alt+4
Next Incomplete	Alt+5

Table 97 Keyboard Equivalents for Data Form and Ad Hoc Grid Management Tasks

Managing Data Form Tasks	Keyboard Equivalent
Manage Data Forms and Ad Hoc Grids	Submenu Items/Buttons
Create (data form folder)	Alt+1
Rename (data form folder)	Alt+5
Assign Access (data form folder)	Alt+2
Move (data form folder)	Alt+3
Delete (data form folder)	Alt+4
Create (data form)	Alt+C
Edit	Alt+I
Move	Alt+M
Delete	Alt+9
Assign Access (data form)	Alt+G
Rename (data form)	Alt+B
Search	Alt+]
Search Up	Alt+M
Search Down	Alt+W
Check All	Alt+;
Assign Access	Submenu Items/Buttons
Add Access	Alt+A
Edit Access	Alt+E
Remove Access	Alt+R

Managing Data Form Tasks	Keyboard Equivalent
Add Access	Submenu Items/Buttons
Add	Alt+A
Migrate Identities	Alt+M
Create/Edit Simple Data Form	Submenu Items/Buttons
Properties	Alt+1
Layout	Alt+2
Other Options	Alt+3
Business Rules	Alt+4
Moving Dimensions	Submenu Items/Buttons
Move dimension to row axis	Ctrl+Shift+R
Move dimension to column axis	Ctrl+Shift+C
Move dimension to Page axis	Ctrl+Shift+P
Move dimension to POV axis	Ctrl+Shift+V
Add Validation Rule	Submenu Items/Buttons
Add Rule	Alt+S
Duplicate	Alt+8
View Rule	Alt+5
Validate	Alt+S
Process Cell	Submenu Items/Buttons
Add	Alt+1
Delete	Alt+2
Copy	Alt+3
Paste	Alt+4
OK	Alt+S
Other Options (create/edit simple data forms)	Submenu Items/Buttons
Add	Alt+I
Remove	Alt+J
Add All	Alt+K

Managing Data Form Tasks	Keyboard Equivalent
Remove All	Alt+L
Business Rules (create/edit simple data forms)	Submenu Items/Buttons
Add	Alt+I
Remove	Alt+J
Add All	Alt+K
Remove All	Alt+L
Properties	Alt+T

Table 98 Keyboard Equivalents for Managing Composite Data Form Tasks

Managing Composite Data Form Tasks	Keyboard Equivalent
Create (composite data form)	Submenu Items/Buttons
Properties	Alt+1
Layout	Alt+2
Business Rules	Alt+3
Layout (composite data form), selecting data forms	Submenu Items/Buttons
Move to Section Properties (add/remove data forms)	Alt+M
Add	Alt+I
Remove	Alt+J
Add All	Alt+K
Remove All	Alt+L
Layout (composite data form), navigation	Submenu Items/Buttons
Move between sections in composite data forms	Up, down, left and right arrow keys
Move from properties pane to the active section	Ctrl+Alt+S

Table 99 Keyboard Equivalents for Managing Task List Tasks

Managing Task List Tasks	Keyboard Equivalent
Create (task list folder)	Alt+1
Rename (task list folder)	Alt+5
Move (task list folder)	Alt+3
Delete (task list folder)	Alt+4

Managing Task List Tasks	Keyboard Equivalent
Assign Access	Alt+G
Check Box	Alt+;
Edit Task List	Submenu Items/Buttons
Task List	Alt+1
Instructions	Alt+2
Edit Task	Submenu Items/Buttons
Task	Alt+1
Property	Alt+2
Assign Access	Submenu Items/Buttons
Add Access	Alt+2
Edit Access	Alt+E
Remove Access	Alt+R
Add/Edit Access	Submenu Items/Buttons
Users	Alt+1
Groups	Alt+2
Migrate Identities	Alt+M
Remove Nonprovisioned Users, Groups	Alt+N

Table 100 Keyboard Equivalents for Smart List, Broadcast Messages and Application Properties Tasks

Smart List, Broadcast Messages, Application Properties Tasks	Keyboard Equivalent
Add Smart Lists	Submenu Items/Buttons
Properties	Alt+1
Entries	Alt+2
Preview	Alt+3
Broadcast Messages	Submenu Items/Buttons
Send	Alt+S
Application Properties	Submenu Items/Buttons
Application Properties	Alt+5
System Properties	Alt+6

Smart List, Broadcast Messages, Application Properties Tasks	Keyboard Equivalent
Add (Application or System Properties)	Alt+2

Table 101 Keyboard Equivalents for Working With Dimensions

Dimension-Related Tasks	Keyboard Equivalent
Dimensions Page	Submenu Items/Buttons
Dimensions	Alt+2
Performance Settings	Alt+3
Evaluation Order	Alt+4
Add/Edit Dimension Member Tabs	Submenu Items/Buttons
Member Properties	Alt+1
UDA	Alt+2
Member Formula	Alt+3
Member Formula	Submenu Items/Buttons
Validate Member Formula	Alt+V
Member Formula Validation Status	Submenu Items/Buttons
Show Details	Alt+S
Dimensions Tab	Submenu Items/Buttons
Select for Dimension	Alt+Q
Add Dimension	Alt+M
Edit Dimension	Alt+G
Sort	Alt+K
Sort Ascending	Alt+U
Sort Descending	Alt+N
Search	Alt+S
Search Up Link	Alt+W
Search Down Link	Alt+R
Show Ancestors	Alt+B
Show Usage	Alt+Y
Custom Attributes	Alt+0 (zero)

Dimension-Related Tasks	Keyboard Equivalent
Options	Alt+P
Add Summary Time Period	Alt+6
DTS	Alt+J
Performance Settings Tab	Submenu Items/Buttons
Select Plan Type	Alt+S
Account, Scenario, Version Dimension	Submenu Items/Buttons
Copy Scenario	Alt+B
Show Usage	Submenu Items/Buttons
Prev	Alt+P
Currency Dimension	Submenu Items/Buttons
Add	Alt+C
Reporting Currency	Alt+R
Employee Entity Dimension	Submenu Items/Buttons
Expand	Right-arrow key
Collapse	Left-arrow key
Custom Attributes	Alt+0 (zero)
Show Ancestors	Alt+B
Manage Attributes (employee dimension)	Submenu Items/Buttons
Create	Alt+C
Modify	Alt+M
Delete	Alt+T
Create	Alt+R
Modify	Alt+O
Delete	Alt+E
Alias	Alt+L
Period Dimension	Submenu Items/Buttons
Add Summary Period	Alt+6
DTS	Alt+J

Dimension-Related Tasks	Keyboard Equivalent
Options	Alt+P
View Period	Alt+5
Year Dimension	Submenu Items/Buttons
Add Years	Alt+C
Edit Year	Alt+I
Options	Alt+P
View Year	Alt+5
Add Year	Submenu Items/Buttons
Add Year	Alt+6
UDA	Alt+2
UDA Tab	Submenu Items/Buttons
Add (link)	Alt+I
Remove (link)	Alt+J
Remove All (links)	Alt+M

Table 102 Keyboard Equivalents for Copying Data and Selecting Dimensions Tasks

Copying Data, Selecting Dimensions Tasks	Keyboard Equivalent
Copy Data	Submenu Items/Buttons
Add Dimensions	Alt+3
Copy Data	Alt+C
Dimension Member Selection	Submenu Items/Buttons
Expand	Right-arrow key
Collapse	Left-arrow key
Search	Alt+3
Search Up	Alt+4
Search Down	Alt+5
Display Properties	Alt+6
Add All	Alt+X
Remove All	Alt+Y

Copying Data, Selecting Dimensions Tasks	Keyboard Equivalent
Keep Only By Function	Alt+7
Keep Only By Attribute	Alt+8
Keep Only By Level or Generation	Alt+9
Add Graphics	Alt+I
Remove Graphics	Alt+J
Remove All Graphics	Alt+Y

Table 103 Keyboard Equivalents for Business Rules Security, Clear Cell Details, Alias Table Tasks

Business Rules Security, Clear Cell Details, Alias Table Tasks	Keyboard Equivalent
Business Rules Security (folders)	Submenu Items/Buttons
Create	Alt+1
Rename	Alt+R
Assign Access	Alt+2
Delete	Alt+4
Clear Cell Details	Submenu Items/Buttons
Add All Dimensions	Alt+3
Add Dimensions	Alt+4
Alias Table	Submenu Items/Buttons
Clear Values	Alt+U

Table 104 Keyboard Equivalents for Reporting and Miscellaneous Administrative Tasks

Reporting, Miscellaneous Administrative Tasks	Keyboard Equivalent
Reporting	Submenu Items/Buttons
Data Forms	Alt+2
Add	Alt+5
Remove	Alt+6
Add All	Alt+7
Remove All	Alt+8
Create Report	Alt+C
Planning Unit Annotations	Alt+3

Reporting, Miscellaneous Administrative Tasks	Keyboard Equivalent
Auditing	Alt+4

Table 105 Keyboard Equivalents for Application-Related Tasks

Application-Related Tasks	Keyboard Equivalent
Application Settings	Submenu Items/Buttons
Register Shared Services	Alt+M
Unregister Shared Services	Alt+W
Select	Alt+4
System Settings	Alt+2
Custom Tools	Alt+3
Register Shared Services	Submenu Items/Buttons
New Application Group	Alt+N
Existing Application Group	Alt+E
Unassigned Application Group	Alt+U
Create Application	Submenu Items/Buttons
Select Tab	Alt+2
Calendar Tab	Alt+3
Currencies Tab	Alt+4
Plan Type Tab	Alt+5
Finish	Alt+6
Log Off	Alt+8
Previous	Alt+P
Delete Application	Submenu Items/Buttons
Delete Application	Alt+9
Register Application	Submenu Items/Buttons
Register Application	Alt+R
Map Reporting Application	Submenu Items/Buttons
New	Alt+C
Push Data	Alt+P

Application-Related Tasks	Keyboard Equivalent
Select Application	Alt+1
Map Dimension	Alt+2
Point of View	Alt+3

Table 106 Keyboard Equivalents for Approvals, Planning Unit Hierarchy Tasks

Planning Unit Hierarchy Tasks	Keyboard Equivalent
Scenario and Version Assignment	Alt+G
Synchronize (planning unit hierarchy)	Alt+8
Create/Edit Planning Unit Hierarchy	Submenu Items/Buttons
Approvals Dimension	Alt+1
Primary and Subhierarchy Selection	Alt+2
Assign Owners	Alt+3
Primary and Subhierarchy Selection	Submenu Items/Buttons
Reset to Default Hierarchy	Alt+R
Assign Owners, Select and Assign Owners	Submenu Items/Buttons
Remove All	Alt+M

Table 107 Keyboard Equivalents for Manage Data Source Tasks

Manage Data Source Tasks	Keyboard Equivalent
Manage Data Source, Create Data Source	Submenu Items/Buttons
Validate Database Connection	Alt+4
Validate Essbase Connection	Alt+5
Edit Data Source	Alt+I
Delete Data Source	Alt+9
OK	Alt+S

Table 108 Keyboard Equivalents for Upgrade Wizard Tasks

Upgrade Wizard Tasks	Keyboard Equivalent
Upgrade Wizard	Submenu Items/Buttons
Update Data Sources Tab	Alt+1
Apply to Selected (Update Relational Information)	Alt+A

Upgrade Wizard Tasks	Keyboard Equivalent
Apply to Selected (Update Essbase Information)	Alt+E
Validate	Alt+V
Upgrade Applications Tab	Alt+2
Upgrade	Alt+U
Update Reporting Essbase Servers Tab	Alt+3
Apply to Selected	Alt+E

Table 109 Keyboard Equivalents for Data Load and Currency Conversion Tasks

Data Load, Conversion Tasks	Keyboard Equivalent
Data Load Administration	Submenu Items/Buttons
Add New Row	Alt+W
Duplicate Row	Alt+P
Remove Row	Alt+R
Create Database	Submenu Items/Buttons
Create (in Warning)	Alt+R
Manage Currency Conversion	Submenu Items/Buttons
Create File Tab	Alt+2
Details Tab	Alt+3
Currency Search	Alt+6
Scenario Search	Alt+7
Version Search	Alt+8
Create New	Alt+W
Manage Exchange Rates	Submenu Items/Buttons
Rate Table	Alt+1
Exchange Rate	Alt+2
Check All	Alt+;
Exchange Rate Table right-click menu	Submenu Items/Buttons
Fill Year	Alt+Y
Fill Table	Alt+T

Data Load, Conversion Tasks	Keyboard Equivalent
Copy	Alt+C
Sort, ascending	Alt+A
Sort, descending	Alt+E

Table 110 Keyboard Equivalents for Data Form Tasks

Data Form Tasks	Keyboard Equivalent
Select current data form	Ctrl+Shift+Space
Select column	Ctrl+Space
Select row	Shift+Space
Minimize column	Alt+9
Restore column	Alt+0 (zero)
Restore all columns	Alt+8
Reset all columns to default width	Alt+7
Lock/Unlock cell	Alt+!
Add, Edit document	Alt+6
Open document	Alt+5
Move focus to Adjust Data, Grid Spread or Mass Allocate dialog	Alt+2
Select Date Selector	Alt+K
Shift between data forms in composite data forms	Alt+]
Edit/View account annotation	Alt+[
Show Context Menu on Row for a Data Form	Shift+Space, Shift+F10
Show Context Menu for the First Header in a Row	Shift+Space, Shift+F10
Show Context Menu for the Second Header in a Row	Shift+Space, Alt+Y
Show Context Menu for a Data Form Cell **Note:** If a data form cell contains a Smart List, navigate to the **Edit** menu to access cell level context menus.	Shift+F10
Arrow Keys to Navigate Between Menu Items	Submenu Items/Buttons
Close the Context Menu	Esc
Move from Data Form to the Page Drop-Down Box	Ctrl+1
Move to Current Data Form	Ctrl+2

Data Form Tasks	Keyboard Equivalent
Expand/Collapse a Row	Space
Data Validation Flyout Image	Ctrl+Shift+D

Table 111 Copy Version and Manage Business Rules Tasks

Copy Version, Manage Business Rules Tasks	Keyboard Equivalent
Copy Version	Submenu Items/Buttons
Add	Alt+3
Add All	Alt+5
Remove	Alt+4
Remove All	Alt+6
Business Rules	Submenu Items/Buttons
Page	Alt+5
Go	Alt+9

Glossary

#MISSING See missing data.

access permissions A set of operations that a user can perform on a resource.

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.

adapter Software that enables a program to integrate with data and metadata from target and source systems.

aggregate function A type of function, such as sum or calculation of an average, that summarizes or performs analysis on data.

aggregate storage database The database storage model designed to support large-scale, sparsely distributed data which is categorized into many, potentially large dimensions. Upper level members and formulas are dynamically calculated, and selected data values are aggregated and stored, typically with improvements in overall aggregation time.

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.

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.

asset account An account type that stores values that represent a company's assets.

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 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.

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.

balance account An account type that stores unsigned values that relate to a particular time.

base currency The currency in which daily business transactions are performed.

base entity An entity at the bottom of the organization structure that does not own other entities.

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.

business rules Logical expressions or formulas that are created within an application to produce a desired set of resulting values.

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.

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. c

calendar User-defined time periods and their relationship to each other. Q1, Q2, Q3, and Q4 comprise a calendar or fiscal year.

CDF See custom-defined function.

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.

child A member with a parent above it in the database outline.

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.

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.

cube A block of data that contains three or more dimensions. An Essbase database is a cube.

currency conversion A process that converts currency values in a database from one currency into another. For example, to convert one U. S. dollar into the European euro, the exchange rate (for example, 0.923702) is multiplied by the dollar (1* 0.923702). After conversion, the European euro amount is .92.

custom dimension A dimension created and defined by users. Channel, product, department, project, or region could be custom dimensions.

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.

data form A grid display that enables users to enter data into the database from an interface such as a Web browser, and to view and analyze data or related text. Certain dimension member values are fixed, giving users a specific view into the data.

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.

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.

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.

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.

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.

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.

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 time series functions Hidden outline members that provide period-to-date reporting at up to eight levels, such as year-to-date and month-to-date totals. An administrator assigns Dynamic Time Series members to members of the time-based dimension in your database outline.

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.

entity 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 run-time). You define the EPM Oracle instance directory location during configuration with EPM System Configurator.

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.

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.

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.

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).

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.

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.

frame An area on the desktop. Two main areas: the navigation and workspace frames.

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.

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.

implied share A member with one or more children but only one that is consolidated, so the parent and child share a value.

iteration A pass of the budget or planning cycle in which the same version of data is revised and promoted.

Java Database Connectivity (JDBC) A client-server communication protocol used by Java-based clients and relational databases. The JDBC interface provides a call-level API for SQL-based database access.

JSP Java Server Page.

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.

line item detail The lowest level of detail in an account.

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 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.

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.

LRO See linked reporting object.

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

MDX (multidimensional expression) A language used for querying and calculation in multidimensional-compliant databases.

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.

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.

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.

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.

online analytical processing (OLAP) A multidimensional, multiuser, client-server computing environment for users who analyze consolidated enterprise data in real time. OLAP systems feature drill-down, data pivoting, complex calculations, trend analysis, and modeling.

outline The database structure of a multidimensional database, including all dimensions, members, tags, types, consolidations, and mathematical relationships. Data is stored in the database according to the structure defined in the outline.

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.

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.

planner A user who can input and submit data, use reports that others create, execute business rules, use task lists, enable e-mail notification for themselves, and use Smart View. 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.

POV (point of view) A feature for setting data focus by selecting members that are not already assigned to row, column, or page axes. For example, selectable POVs in FDM could include location, period, category, and target category. In another example, using POV as a filter in Smart View, you could assign the Currency dimension to the POV and select the Euro member. Selecting this POV in data forms displays data in Euro values.

precision Number of decimal places displayed in numbers.

provisioning The process of granting users and groups specific access permissions to resources.

relational database A type of database that stores data in related two-dimensional tables. Contrast with multidimensional database.

reporting currency The currency used to prepare financial statements, and converted from local currencies to reporting currencies.

role The means by which access permissions are granted to users and groups for resources.

root member The highest member in a dimension branch.

runtime prompt A variable that users enter or select before a business rule is run.

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.

shared member A member that shares storage space with another member of the same name, preventing duplicate calculation of members that occur multiple times in an Essbase outline.

Shared Services Registry The part of the Shared Services repository that manages EPM System deployment information for most EPM System products, including installation directories, database settings, computer names, ports, servers, URLs, and dependent service data.

sibling A child member at the same generation as another child member and having the same immediate parent. For example, the members Florida and New York are children of East and each other's siblings.

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.

Structured Query Language A language used to process instructions to relational databases.

supporting detail Calculations and assumptions from which the values of cells are derived.

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.

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.

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.

upgrade The process of deploying a new software release and moving applications, data, and provisioning information from an earlier deployment to the new deployment.

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 data 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.

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.

variance The difference between two values (for example, between planned and actual values).

varying attribute An attribute association that changes over one or more dimensions. It can be used to track a value in relation to these dimensions; for example, the varying attribute Sales Representative, associated with the Product dimension, can be used to track the value Customer Sales of several different sales representatives in relation to the Time dimension. Varying attributes can also be used for member selection, such as finding the Products that a Sales Representative was responsible for in May.

version 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.

Index