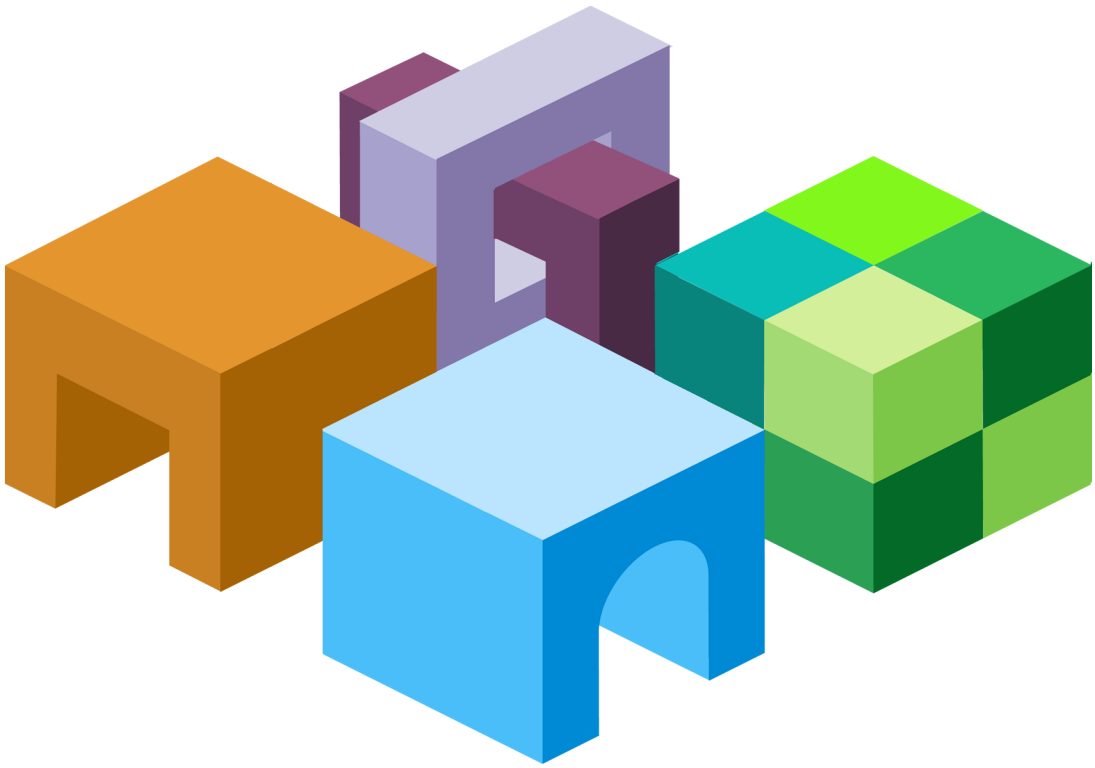


# ORACLE® HYPERION PROFITABILITY AND COST MANAGEMENT, FUSION EDITION

*RELEASE 11.1.1.3*

## ADMINISTRATOR'S GUIDE

ORACLE®
ENTERPRISE PERFORMANCE MANAGEMENT SYSTEM

Profitability and Cost Management Administrator's Guide, 11.1.1.3



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.

## TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, seven days a week. For TTY support, call 800.446.2398.

# 1 Overview

**In This Chapter**



Oracle Hyperion Profitability and Cost Management, Fusion Edition is an analytical application, that is accessed from Oracle Enterprise Performance Management Workspace, Fusion Edition, and used to accurately measure, allocate, and manage costs and revenues, compute profitability for business segments, and measure profitability by using cost decomposition, consumption-based costing, and scenario playing.

Profitability and Cost Management is an integral part of EPM Workspace, using various components to build and manage its applications:

- Use EPM Workspace to access Profitability and Cost Management, and manage other components to build the application, control security, and product reports.
- Use Oracle's Hyperion® Shared Services to create and manage user accounts, including the definition of security roles to determine which models users can access.
- Use Oracle Hyperion EPM Architect, Fusion Edition to build and maintain both Shared and Local dimensions and members for use in Profitability and Cost Management. Through the Dimension Library, you can build the application using common dimensions and members that already exist in other applications, such as Oracle Hyperion Planning, Fusion Edition.

  Using Performance Management Architect, dimensions can exist in an Application either as Shared or Local:

  - Shared Dimensions reside in the Shared Library in Performance Management Architect, and can be used by multiple applications.
  - Local Dimensions are detached, independent dimensions that only exist in one application, such as Profitability and Cost Management. These dimensions are used only for the application for which they have been created.
- Use Oracle Essbase to create the application outline, and store and calculate the assignments in Profitability and Cost Management.

- Use Oracle Hyperion Enterprise Performance Management System Lifecycle Management to migrate an application, multidimensional database, repository, or individual artifacts across product environments and operating systems.
- Create reports of the calculated results, using Oracle's Hyperion Reporting and Analysis, Oracle Hyperion Financial Reporting, Fusion Edition, Oracle's Hyperion® Web Analysis, or third party products, such as Microsoft Excel.

# Accessibility in Profitability and Cost Management

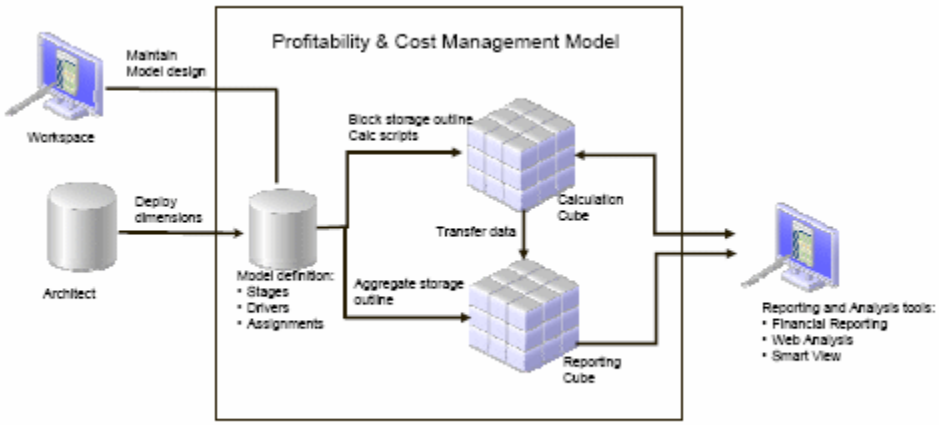
Profitability and Cost Management provides keyboard shortcuts for the main functions.

The Accessibility features are documented in the Accessibility Appendix in the *Oracle Hyperion Profitability and Cost Management User's Guide*. These features are unique to Profitability and Cost Management.

# Understanding the Product Architecture

Accessed through EPM Workspace, Profitability and Cost Management is an analytical application that resides on top of Essbase. The application enables business users to model their business for profitability and cost management, and use that model information to create Essbase databases that can be fine-tuned for profitability and cost analysis without having to understand a scripting language.

**Figure 1 Profitability and Cost Management Product Architecture**




Profitability and Cost Management leverages Performance Management Architect and Shared Services for the centralized management of application metadata and security.

Application administrators create the Profitability and Cost Management dimensions using Performance Management Architect. User access is managed centrally with Shared Services. When the dimension metadata is ready, it is deployed to a Profitability and Cost Management

application, or model. The dimensions in Performance Management Architect can be shared by multiple models.

The model design contains the information needed to generate the Essbase outline and calculation script required by the Essbase component of the model. Each model requires access to the following databases:

- Relational database that stores the model design, including the dimension metadata deployed from Performance Management Architect
- Essbase database that includes a Calculation database (BSO) and a Reporting database (ASO).

**Note:** Only one database is required to store multiple models.

Results from the Reporting database may be viewed in reporting and analysis tools, such as Oracle Hyperion Financial Reporting, Fusion Edition, Oracle's Hyperion® Web Analysis, and Oracle Hyperion Smart View for Office, Fusion Edition.

# Administrative Tasks

The Profitability and Cost Management Administrator or *admin* role enables you to perform these tasks:

- Create and manage user accounts using Shared Services.
- Provision users for Shared Services authentication.
- Build the model database using Performance Management Architect to select the common dimensions and members.
- Create staging tables in the source database to import model data from relational databases into Profitability and Cost Management.
- Generate multidimensional Essbase databases.
- Create, update and delete model stages, drivers and points of view (POVs).
- Create, update and delete driver selections and assignments.
- Create, update and delete applications and application preferences.
- View and modify model data.
- View trace allocations.
- Back up and restore Profitability and Cost Management model components.
- Transfer applications from one environment to another using the Lifecycle Management Utility. Promote data from one environment, such as development or testing, to another environment, such as production.
- Monitor changes made to business objects.

The Profitability and Cost Management Administrator also manages the model metadata and data through the EPM Workspace. See Chapter 4, “Managing Applications and Dimensions Using Performance Management Architect”.

# Accessing Profitability and Cost Management After Installation

Profitability and Cost Management is an integral part of EPM Workspace, and uses common applications to manage the application and security. After installation, you must perform a number of steps to create the first Profitability and Cost Management application. After the application is created, you need to import data into Profitability and Cost Management.

➤ To create the first Profitability and Cost Management application after installation:

1 **Ensure the following products are installed, configured and running:**

- EPM Workspace
- Shared Services
- Performance Management Architect
- Essbase
- Profitability and Cost Management

**Note:** This list represents the minimum installation required to use Profitability and Cost Management; however, you may install additional products at any time.

If you need to determine which products have already been installed

For complete installation instructions, see the *Oracle Hyperion Enterprise Performance Management System Installation and Configuration Guide* and the *Oracle Hyperion Enterprise Performance Management System Installation Start Here*.

2 **On the EPM Workspacemain menu, select the Navigate icon  , then Administer, and then Dimension Library to access Performance Management Architectto create the dimensions required for the first application.**

For detailed instructions on creating dimensions, see the *Oracle Hyperion Enterprise Performance Management Architect Administrator's Guide*. For information on required dimensions, see Chapter 4, "Managing Applications and Dimensions Using Performance Management Architect".

3 **After the dimensions have been selected for Profitability and Cost Management, select Navigate, then Administer, and then Application Library.**

4 **From the Application Library, select File, then New, and then Application Wizard to create a new Profitability and Cost Managementapplication. For detailed instructions on creating a new application, see the *Oracle Hyperion Enterprise Performance Management Architect Administrator's Guide*.**

5 **From the Application Library, right-click the new application name, and select Deploy.**

The application name and selected dimensions are forwarded to the Profitability and Cost Managementserver. For detailed instructions, see the *Oracle Hyperion Enterprise Performance Management Architect Administrator's Guide*.

6 **From EPM Workspace, select Navigate, then Administer, and then User Management to register the new application with Shared Services. If required, you can also add users and security roles at this time. For**

detailed instructions, see the *Oracle Hyperion Enterprise Performance Management System Security Administration Guide*.

7 On the EPM Workspacemain menu, click the Navigate icon  .

8 From Navigate, select **Applications**, then **Profitability**, and then select the new application.

**Note:** If the newly created application is not found, select Refresh, and then repeat step 8 to select the newly created application.

9 From **Task Areas**, select **Manage Models**, then **Import Staging Tables** to import the data you require to begin building a model. See Chapter 5, "Importing Data in Profitability and Cost Management".

# Launching Profitability and Cost Management

Profitability and Cost Management can only be accessed through EPM Workspace.

➤ To access Profitability and Cost Management:

1 Ensure the following applications have been configured, and are running:

- EPM Workspace
- Shared Services
- Performance Management Architect
- Essbase
- Profitability and Cost Management

See the *Hyperion EPM System Installation and Configuration Guide* and the *Oracle Hyperion Enterprise Performance Management System Installation Start Here* for instructions.

2 In your Web browser, access the EPM Workspace Web page.

By default, the URL is `http://server name:19000/workspace/`.

3 Enter the EPM Workspace user name and password.

**Note:** Both the user name and password are case-sensitive.

4 Click **Log On**.

The main EPM Workspace page is displayed.

5 On the EPM Workspace main menu, click the Navigate button  .

6 From Navigate, select **Applications**, then **Profitability** and then select the application you want to view.

# 2 Managing Security and Authorizing Users

**In This Chapter**



## About User Setup and Provisioning

Before working with Profitability and Cost Management, the Administrator must set up users and groups, and assign the appropriate security role to each one. The authorization provided for each security role determines which functions and data a user or group may access. During configuration, select Shared Services as the authentication mode, as described in the *Oracle Hyperion Enterprise Performance Management System Installation and Configuration Guide*.

You create, maintain and provision users and groups for Profitability and Cost Management through EPM Workspace. The provisioning process requires you to have both Shared Services and Profitability and Cost Management configured and running. External authentication ensures that the applications communicate seamlessly to provision users easily and accurately.

The following steps provide an overview of the process to set up and provision users and groups:

1. From the Navigate icon  in EPM Workspace, select **Administer**, then **User Management** to access the Shared Services screens.
2. Create users. The first time that you log on, an administrator (*admin*) user is automatically created for your product.
3. Provision the users with the appropriate security role, and access to the required projects and applications.

   **Caution!** If the user needs to access EPM Workspace to perform any tasks outside of Profitability and Cost Management, you must provision the selected user with the EPM Workspace role as well.

4. Create groups, as required. You can provision groups within groups.
5. Log on to EPM Workspace, and open Profitability and Cost Management.
6. Verify that the user can log on to Profitability and Cost Management, and is able to see the provisioned projects and applications.

For detailed instructions on setting up and provisioning users and groups, see the *Oracle Hyperion Enterprise Performance Management System Security Administration Guide*.

## Assigning Security Roles

In Profitability and Cost Management, each user ID is assigned a security role:

- Administrator (*admin* is the default security role when you log on to Shared Services)
- Power User
- Interactive User
- View User

The assigned security role determines the level of access or privileges available for that user. A user can be granted multiples roles, and at the time an action is initiated, the permission is checked for that specific action.

When an access level is assigned to a group of users, similar security access is granted to all members of that group. Depending on the access requirements for a particular user, the assigned security may be modified to attach a wider or narrower access. For example, a View User assigned to a group that has Power User security authorization assumes that higher level of security.

**Note:** A user must exist and have an assigned security role before you can assign the user to a group.

See the *Oracle Hyperion Enterprise Performance Management System Security Administration Guide* for detailed instructions.

## Profitability and Cost Management Security Roles

The security roles in Table 1 are specific to Profitability and Cost Management.

**Caution!** If the user requires access to other products, such as an Application Creator in EPM Workspace or Dimension Editor in Performance Management Architect, those additional security roles must be assigned separately. See the *Oracle Hyperion Enterprise Performance Management System Security Administration Guide*.

**Table 1** Profitability and Cost Management Security Roles

| Security Role | Type of Role | Description |
|---|---|---|
| Administrator (*admin*) | Power | • Create and maintain user accounts and security roles, and provision users, using Oracle's Hyperion® Shared Services<br>• Generate Essbase databases |

| Security Role | Type of Role | Description |
|---|---|---|
| | | • Set up and maintain application preferences.<br>• Build the model database using Performance Management Architect to select the common dimensions and members.<br>• Create and maintain elements within the model, such as stages, drivers, POVs, driver associations, assignments, and application preferences.<br>• Create staging tables in the source database to import model data and metadata from relational databases into Profitability and Cost Management.<br>• Perform POV Copy, calculation, validation, data entry and trace allocations.<br>• Deploy to Essbase and generate calculation scripts.<br>• Import and export data<br>• Use the Lifecycle Management Utility to promote data from one environment, such as development or testing, to another environment, such as production.<br>• Back up and restore Profitability and Cost Management model components.<br>• Monitor changes made to business objects. |
| Power User | Power | • Create and maintain elements within the model, such as stages, drivers, POVs, driver associations, assignments and application preferences.<br>• Perform POV Copy, calculation, validation, data entry and trace allocations.<br>• Deploy to Essbase and generate calculation scripts.<br>• Import and export data |
| Interactive User | Interactive | • View all modelling screens<br>• View and modify data in the Data Entry screen |

| Security Role | Type of Role | Description |
|---|---|---|
| View User | Interactive | View only access for these functions:<br>● Trace Allocations<br>● Application Preferences<br>● Model Stages, Drivers and POVs |

For a complete description of all security roles, see the *Oracle Hyperion Enterprise Performance Management System Security Administration Guide*.

# Auditing Changes in Profitability and Cost Management

You can monitor activity and changes in your application using the Audit feature in the Oracle's Hyperion® Shared Services Console, and then generate audit reports detailing the results.

There are three types of audit reports available:

- Security reports
- Artifact Reports
- Config Reports

The audit reports contain activity details for the selected audit area, including the following information:

- Date
- Application
- User
- Artifact type and name
- Task that was performed

Auditing must be enabled before you can generate any reports. See "Enabling the Audit Feature" on page 18. These reports can be exported as .CSV files. For instructions on generating audit reports, see the *Oracle Hyperion Enterprise Performance Management System Security Administration Guide*

## Enabling the Audit Feature

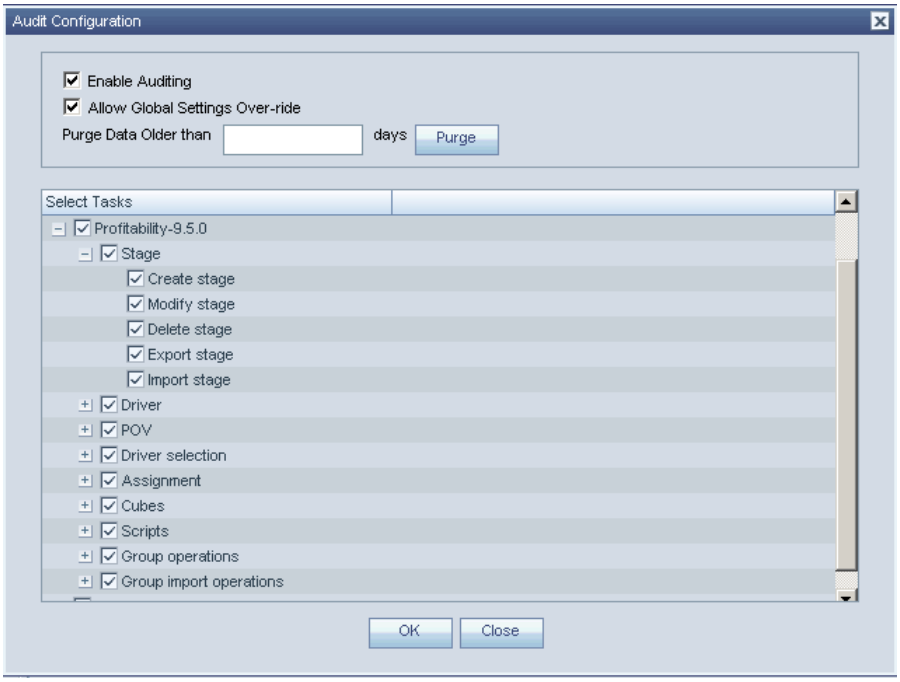
➤ To enable auditing:

1 From **Navigate**  in EPM Workspace, select **Administer**, then **Shared Services Console.**

The Shared Services Console is displayed.

2 Select **Administration**, then **Configure Auditing**.

The Audit Configuration screen is displayed.




3 Select **Enable Auditing**.

The Allow Global Settings Over-ride and Select Tasks list are activated.

4 From the **Select Tasks** list, select the areas of the application to be enabled for audit. You can select an entire area, or expand each option to choose separate steps to monitor.

**Table 2** Profitability and Cost Management Audit Tasks

| Monitored Area | Available Monitored Tasks |
|---|---|
| Stage | • Create stage<br>• Modify stage<br>• Delete stage<br>• Export stage<br>• Import stage |
| Driver | • Create driver<br>• Modify driver<br>• Delete driver<br>• Export driver<br>• Import driver |
| POV | • Create POV<br>• Modify POV<br>• Delete POV<br>• Export POV<br>• Import POV |
| Driver Selection | • Create driver selection |

| Monitored Area | Available Monitored Tasks |
|---|---|
| | • Modify driver selection<br>• Delete driver selection |
| Assignment | • Create assignment<br>• Modify assignment<br>• Delete assignment |
| Assignment Rules | • Create assignment rule<br>• Modify assignment rule<br>• Delete assignment rule<br>• Create rule selection<br>• Remove rule selection |
| Cubes | • Deploy calculation cube<br>• Deploy reporting cube<br>• Transfer data |
| Genealogy | • Calculate genealogy |
| Scripts | • Deploy allocation script<br>• Deploy genealogy script<br>• Execute allocation script<br>• Execute genealogy script |
| Group operations | • Copy assignments<br>• Delete assignments<br>• Copy driver selections<br>• Delete driver selection rules<br>• Delete driver selection exceptions |
| Group import operations | • Import stages<br>• Import POVs<br>• Import drivers<br>• Import driver selections<br>• Import assignments |
| Group export operations | • Export stages<br>• Export POVs<br>• Export drivers<br>• Export driver selections<br>• Export assignments |

5 **Click OK.**

A message is displayed to confirm the audit configuration has been saved.

6 **Click Yes.**

7 **Optional:** Generate audit reports to review audit results, as outlined in the *Oracle Hyperion Enterprise Performance Management System Security Administration Guide*.

# 3 Working with Profitability and Cost Management Dimensions and Members

**In This Chapter**



## About Dimensions and Metadata

Dimensions and metadata are created and maintained by the Profitability and Cost Management Administrator, using Performance Management Architect. By using common dimensions and members, Profitability and Cost Management can easily use and transfer common data with other products, such as Planning, saving time and effort, and improving accuracy. For detailed information on creating and maintaining dimensions and metadata, see the *Oracle Hyperion Enterprise Performance Management Architect Administrator's Guide*.

Profitability and Cost Management uses the dimensions and members created in Performance Management Architect to represent many of the structural elements of the business model:

- Dimensions for each model stage
- Time Periods
- Point of View (POV)
- Other attribute dimensions, as required.

For each dimension, both a dimension type and dimension name must be specified:

- Dimension type is a dimension property that enables the use of predefined functionality for selected dimensions. See "Dimension Types" on page 24.
- Dimension name is assigned to identify the contents of the dimension, in relation to your organization or business. For example, a dimension of Account type may be given a dimension name, such as General Ledger or Chart of Accounts. The dimension name does not need to reflect the dimension type, although it may.

Using Performance Management Architect, dimensions can exist in an Application either as Shared or Local:

- Shared Dimensions reside in the Shared Library in Performance Management Architect, and can be used by multiple products and applications.
- Local Dimensions are detached, independent dimensions that only exist in one application instance, such as an application of type Profitability and Cost Management. These dimensions are used only within that application for which they have been created, and are not visible or usable even for another application of the same type.

Through Performance Management Architect, you select dimensions and members that exist in other products, or create new dimensions and members specifically for the model. After the dimensions and members are selected for the Profitability and Cost Management model, they are automatically available in the application.

Both the system-generated and user-defined dimensions and members must exist in Performance Management Architect. Dimensions may contain alphanumeric characters, or calculated values.

**Caution!** Although there is no physical limit to the number of dimensions and members that can be created, performance issues occur with large dimensional structures.

The sort order for dimensions in a Profitability and Cost Management model must be set in a certain sequence, to maximize processing and calculation. See "Dimension Sort Order " on page 34.

For detailed instructions on creating and maintaining the dimensions and members, see the *Oracle Hyperion Enterprise Performance Management Architect Administrator's Guide* and the *Oracle Essbase Database Administrator's Guide*.

See these sections for information about the Profitability and Cost Management dimensions:


- "Dimension Types" on page 24
- "Measure Dimensions" on page 25
- "AllocationType Dimension " on page 32
- "Business Dimensions" on page 33
- "Attribute Dimensions" on page 34
- "Dimension Sort Order " on page 34


# Dimension Types

For Profitability and Cost Management, these types of dimensions are available in the Essbase outline:

- System dimensions, such as the Measures and AllocationType dimensions, must be created based on the requirements listed in "Measure Dimensions" on page 25 and "AllocationType Dimension " on page 32.

- Business dimensions are created to describe the business elements within the model, such as business-specific departments, general ledger accounts, activities, customers, or products. See "Business Dimensions" on page 33.
- Attribute dimensions enable analysis based on the attributes or qualities of dimension members. Attributes describe characteristics of data, such as the size or color of products.
- POV dimensions indicate a specific point of view or version of the model, such as year, scenario, or period. At least one POV dimension is required for each model. You can also create a Version dimension type, to create POV versions that enable you to maintain separate versions of the same POV to monitor the impact of changes to the model, or track different versions of the same model.

**Note:** When defining dimensional outlines, there are restricted characters that may not be used for naming. Oracle strongly suggests that you review the Essbase naming conventions in the *Oracle Essbase Database Administrator's Guide*.

# Measure Dimensions

The Measure dimension contains the dimensions and members required to build, validate and calculate a model.

The Measure dimension is created in Performance Management Architect in one of two ways:

- The user creates the Profitability and Cost Management application using the application wizard, and selects the option to "Auto Create Local Dimensions. The system automatically creates a Measure type dimension.
- The user creates the application manually, by selecting "Create Blank Application" from the application wizard. The user must create their own dimensions and select the Measure dimension type.

In addition to members required for the Measure dimension, users may add any user-defined driver measures (or sub-hierarchies) to the hierarchy under the member 'UserDefinedDriverMeasures'. These dimension members can be modified by the user.

Within the Measure dimension, the types of measures can be grouped into these categories:

- Driver Measures are used in the creation of driver types and formulas. See "Driver Measures" on page 26

  **Note:** In addition to the listed driver measures, if the model contains rules-based assignments, some system members are displayed in the DriverMeasures hierarchy (for example, SysVar1, SysVar2, and so on). Although they are displayed in the hierarchy, these system members are used for internal calculations only, and should not be used for reporting.
- Cost Layer Allocation Measures are used to control allocation of calculated and input costs. See "Cost Layer Allocation Measures" on page 27.
- Revenue Layer Allocation Measures are used to control the allocation of calculated and input revenue. See "Revenue Layer Allocation Measures" on page 29.

- Reporting Measures are used to generate reports, using the calculated and input values to generate total costs and revenue for the model. All the non-level-0 Reporting Measures are calculated. See "Reporting Measures" on page 31.

## Driver Measures

**Table 3** Driver Measures

| Member Name | Alias | Description | Calculated or Input |
|---|---|---|---|
| FixedDriverValue | FixedDV | Default measure for use with driver types that require a fixed driver value parameter | Input |
| Rate | Rate | Default measure for use with driver types that require a rate parameter | Input |
| Quantity | Qty | Default measure for use with driver types that require a quantity parameter | Input |
| Weight | Weight | Default measure for use with driver types that require a weight parameter | Input |
| Percentage | Percent | Default measure for use with a percentage driver type | Input |
| CalculatedDriverValue | CDV | Measure that is the result of the driver formula used in an assignment | Calculated |
| TotalDriverValue | TDV | Measure that is used as the denominator in the allocation formula DV/TDV | Calculated |
| OverrideTotalDriverValue | OvrdTDV | User-entered value that overrides the TotalDriverValue measure as the denominator in allocations | Input |
| TotalDriverValueAfterReciprocals | TDVAftRcp | Measure that is used as the denominator in allocation formulas for non-reciprocal intrastage and post-stage allocations, when a source has been involved in a reciprocal assignment | Calculated |

| Member Name | Alias | Description | Calculated or Input |
|---|---|---|---|
| IdleDriverValue | IdleDV | Measure that is used as the driver value (DV) for calculating IdleCost | Calculated |
| EffectiveTotalDriverValue | EffTDV | Measure that is used to store the Effective Driver Total for drivers that have the “Allow idle” box checked when the driver is defined. | Calculated |
| UserDefinedDriverMeasures | N/A | The member UserDefinedDriverMeasures is where the application-specific, user-defined driver measures are stored.<br>Set the ASOMember DataStorage and BSOMember DataStorage properties as follows:<br>– If this member does not have any children, set to **StoreData**.<br>– If members are added as children to this member, and all these children have the consolidation symbols of IGNORE, then set to **LabelOnly**.<br>**Note:** All driver measures must be unique in the outline. Do not use the name of any existing driver measure in any dimension in the outline as the name of another member (including system, POV and business dimensions) ; otherwise, the Data Entry screen will not properly display the values. | N/A |

# Cost Layer Allocation Measures

**Table 4** Cost Layer Allocation Measures

| Member Name | Alias | Description | Calculated or Input |
|---|---|---|---|
| UnassignedCost | UnAsgCost | Remainder of cost at a source intersection after all assignment and idle | Calculated |

| Member Name | Alias | Description | Calculated or Input |
|---|---|---|---|
| | | calculations have been completed. | |
| CostAssigned | CostAsg | Total cost assigned from a source to post-stage destinations and non-reciprocal intrastage destinations | Calculated |
| CostAssignedIntraStage | CostAsgInt | Sum of costs assigned to intrastage destinations, excluding reciprocal destinations | Calculated |
| CostAssignedPostStage | CostAsgPost | Sum of costs assigned to post-stage destinations | Calculated |
| IdleCost | IdleCost | Idle cost generated by the allocation formula:`IdleDriverValue/OverrideTotalDriverValue` | Calculated |
| NetCostForAssignment | NetCostAsg | Total cost available for assignment after accounting for all prior stage, intrastage and reciprocal assignments | Calculated |
| GrossReceivedCost | GrRecCost | Sum of all costs assigned from prior stages and intrastage assignments, excluding reciprocals and user input costs | Calculated |
| CostInput | CostInput | User-entered cost value for the intersection | Input |
| CostReceived | CostRec | Sum of all costs assigned to an intersection by prior stage and intrastage assignments, excluding costs resulting from reciprocal assignments | Calculated |
| CostReceivedPriorStage | CostRecPri | Sum of costs received on assignments from a prior stage | Calculated |
| CostReceivedIntraStage | CostRecInt | Sum of all costs received on intrastage assignments, excluding reciprocal assignments | Calculated |
| NetReciprocalCost | NetRcpCost | Net effect of a reciprocal assignment on the amount available for assignment to | Calculated |

| Member Name | Alias | Description | Calculated or Input |
|---|---|---|---|
| | | post-stage and non-reciprocal intrastage destinations | |
| ReciprocalCostAssigned | RcpCostAsg | Total cost assigned to reciprocal destination, excluding costs received from the reciprocal | Calculated |
| ReciprocalCostReceived | RcpCostRec | Total cost received from a reciprocal destination | Calculated |
| ReciprocalIntermediate Cost | RcpIntCost | Intermediate value calculated for an intersection after simultaneous equations have been applied, but before reciprocal adjustments have been made | Calculated |
| CostPerDrvUnit | Cost Per Driver Unit | This measure is a child of AllocationMeasures.<br>The formula uses the assigned cost (CostAssigned) divided by the sum of all driver values (TotalDriverValue) to calculate the cost for each unit of driver value. | Calculated |
| UnitCost | Unit Cost. | This measure is a child of AllocationMeasures.<br>The formula uses the amount of cost at a source intersection (NetCostForAssignment) divided by a user-input quantity to calculate the cost per unit. | Calculated |

# Revenue Layer Allocation Measures

**Table 5** Revenue Layer Allocation Measures

| Member Name | Alias | Description | Calculated/Input |
|---|---|---|---|
| UnassignedRevenue | UnAsgRev | Remainder of revenue at a source intersection after all assignment and idle calculations have been completed | Calculated |

| Member Name | Alias | Description | Calculated/Input |
|---|---|---|---|
| RevenueAssigned | RevAsg | Total Revenue assigned from a source to post-stage and non-reciprocal intrastage destinations | Calculated |
| RevenueAssignedIntraStage | RevAsgInt | Sum of revenue assigned to intrastage destinations, excluding reciprocal destinations | Calculated |
| RevenueAssignedPostStage | RevAsgPos | Sum of revenue assigned to post-stage destinations | Calculated |
| IdleRevenue | IdleRev | Idle revenue generated by the allocation formula:`IdleDriverValue/OverrideTotalDriverValue` | Calculated |
| NetRevenueForAssignment | NetRevAsg | Total revenue available for assignment after accounting for all prior stage, intrastage and reciprocal assignments. | Calculated |
| GrossReceivedRevenue | FrRecRev | Sum of all revenue assigned from prior stages and intrastage assignments, excluding reciprocal assignments and user input revenue | Calculated |
| RevenueInput | RevInput | User-entered revenue values for the intersection.<br>Define and store Revenue categories as a hierarchy under 'RevenueInput'. | Input |
| RevenueReceived | RevRec | Sum of all revenue assigned to an intersection by prior stage and intrastage assignments, excluding revenue results from reciprocal assignments | Calculated |
| RevenueReceivedPriorStage | RevRecPri | Sum of revenue received on assignments from a prior stage | Calculated |
| RevenueReceivedIntraStage | RecRecInt | Sum of all revenue received on intra-stage assignments, excluding reciprocal assignments | Calculated |

| Member Name | Alias | Description | Calculated/Input |
|---|---|---|---|
| NetReciprocalRevenue | NetRcpRev | Net effect of a reciprocal assignment on the amount of revenue available for assignment to post-stage destinations and non-reciprocal intrastage destinations | Calculated |
| ReciprocalRevenue Assigned | RcpRevRec | Total revenue assigned to reciprocal destinations, but excluding revenue received from the reciprocal assignment | Calculated |
| ReciprocalRevenue Received | RcpRevAsg | Total revenue received from a reciprocal destination | Calculated |
| ReciprocalIntermediateRevenue | RcpIntRev | Intermediate value calculated for an intersection after simultaneous equations have been applied, but before reciprocal adjustments have been made | Calculated |

# Reporting Measures

**Table 6** Reporting Measures

| Member Name | Alias | Description | Calculated or Input |
|---|---|---|---|
| Profit | Profit | Value of the calculated profit for the selected intersection. This value is the result of the calculation:`NetRevenueForAssignment – NetCostForAssignment` | Calculated |
| GrossCost | GrossCost | Total cost for an intersection, including all possible inputs:<br>● Input values<br>● Prior stage assignments<br>● Intrastage assignments including reciprocals. | |

| Member Name | Alias | Description | Calculated or Input |
|---|---|---|---|
| | | This calculation represents the true total cost of the intersection. | |
| InitialCost | InitialCost | Cost of an intersection before intrastage or reciprocal costs are calculated, including input costs and cost received on assignments from prior stages. | Calculated |
| NetCostAfterIntraStage | NetCostAftInt | Cost of an intersection, including all intrastage cost assignments | Calculated |
| GrossRevenue | GrossRev | Total revenue for an intersection, including all possible inputs:<br>● Input values<br>● Prior stage assignments<br>● Intrastage assignments including reciprocals.<br>This calculation represents the true total revenue for the intersection. | |
| InitialRevenue | InitialRev | Revenue for an intersection before intrastage or reciprocal revenue is calculated, including input revenue, and revenue received on assignments from prior stages. | Calculated |
| NetRevenueAfterIntra Stage | NetRevAftInt | Revenue for an intersection after accounting for all types of intrastage revenue assignments | Calculated |

# AllocationType Dimension

The AllocationType dimension is used to store both direct allocations and allocation genealogy data. When you create reports, the AllocationType dimension enables you to specify which type of allocation data to retrieve.

**Note:** The AllocationType dimension can be renamed, if required.

The system automatically generates the AllocationType dimension if the user creates an application using the wizard, and checks "Auto Create Local Dimensions." If the user selects "Create Blank Application," the user must create their own dimensions and select the AllocationType dimension type.

The data in these dimensions cannot be modified, and it is not visible in Performance Management Architect or Profitability and Cost Management.

In the Essbase outline created by Profitability and Cost Management, the AllocationType dimension contains the following members:

- `AllAllocations` contains the following child members:
  - `DirectAllocation` stores calculated data that has been directly allocated within the model, between a specified source intersection and destination intersection. Direct allocation must be defined as an assignment by the user.
  - `GenealogyAllocation` stores the allocation genealogy that is calculated on indirect links between various concerned intersections in the model.

    Genealogy allocation is not directly defined by the user, but it exists because of two or more direct allocations. For example, A-B-C allocation genealogy data exists because there is a direct allocation from A to B (A-B) and B to C (B-C).
- `SysAllocVar1` stores the value for intrastage assignments on the virtual link, which is part of the value allocated to another node on the same stage, and already available in the `DirectAllocation` member.
- `SysAllocVar2` is used to obtain a sum of the source links of `DirectAllocation`, `Genealogy Allocation` and `SysAllocVar1`.

**Caution!** Do not edit the system members in this dimension, as any modifications may result in the loss of data or the corruption of your model.

# Business Dimensions

Business dimensions describe the business-specific objects within each stage in the model, such as products, customers, regions, employees, and so on. These dimensions and members are created in Performance Management Architect.

Business dimensions may use some or all of the following dimension types, and may apply to one or more stages or models:

- Generic
- Account
- Entity
- Country

When the Essbase outlines are deployed, the business dimensions are created in the Profitability and Cost Management application as basic or generic dimensions, with no type. This feature

enables Profitability and Cost Management to re-use the dimension member and hierarchies that were defined for other applications, such as Planning.

**Note:** This dimension type does not apply to aggregate storage outlines.

When creating a business dimension, the following requirements apply:

- The following properties for the Gen1 member of the dimension must be set to `LABEL_ONLY`:
  - DataStorage(BSO)
  - DataStorage(ASO)
- The first Gen2 child under the Gen1 dimension name is usually set to an ALL member. For example, `AllDepartments` for the Departments dimension.

  The primary hierarchy is hosted under the first Gen2 child. Only the first Gen2 hierarchy is used in allocation modeling, and this hierarchy cannot contain any shared members.
- Additional Gen2 members can host alternate hierarchies, but these hierarchies are not used in allocation modeling. If the dimension is going to host alternate hierarchies, set the DimensionHierarchyType to 'Enabled.'

  These alternate hierarchies are not visible in Profitability and Cost Management modeling screens, and can only be viewed in Essbase.
- A `NoMember` member is required. The last Gen2 child in the hierarchy must always be '`NoMember`', with consolidation set to `IGNORE (~)`.

  **Note:** A `NoMember` member does not have to be created for .ads files because it is added automatically by Performance Management Architect. Do not delete this member.

# Attribute Dimensions

User-defined attribute dimensions (UDA) describe characteristics of data, such as the size or color of products.

You can use these attributes to analyze data, based on the attributes or qualities of dimension members. The attribute dimensions are also used for filtering destination intersections when creating assignment rules.

# Dimension Sort Order

The Dimension Sort Order property controls the order of dimensions in the Essbase outline that is generated by Profitability and Cost Management. The dimension sort order must be set on all dimensions within a model, except Alias and UDA.

**Caution!** If the sort order for any dimension is left blank, the validation will fail.

The Dimension Sort Order property is set in Performance Management Architect, and passed on to Profitability and Cost Management during deployment for use in generating the Essbase outline. For instructions, see "Setting the Dimension Sort Order" on page 35.

The Dimension Sort Order settings for the model are validated in Performance Management Architect. See "Validating and Deploying" on page 56.

## Dimension Sort Order Restrictions

Set the dimension sort using the following restrictions:

- A dimension sort order must be set for every dimension in the model, except Alias and UDA.

  **Note:** The Alias and UDA dimensions are ignored for Dimension Sort Order, as they do not exist as dimensions in Profitability and Cost Management and Essbase.

- The dimension sort order must be sequential.
- Measure dimension is set to 1, by default.
- AllocationType dimension is set to 2, by default.
- Business and POV dimensions must be set to 3 or higher.
- Attribute dimensions must always be sorted as the last dimensions. For example, if you have four attribute dimensions in a sequence of 12 dimensions, the attribute dimensions must be set as 9, 10, 11, and 12.

## Setting the Dimension Sort Order

The processing order for every dimension in the model must be set at the dimension-level using the Dimension Sort Order property. The Dimension Sort Order restrictions must be met; otherwise, validation of the model will fail. For a complete list of restrictions, see "Dimension Sort Order Restrictions" on page 35.

➤ To set the Dimension Sort Order:

1. From EPM Workspace, select **Navigate**, then **Administer**, and then **Dimension Library** to display the Shared Library.
2. In the Property Grid, select **Profitability**.
3. Select the following dimensions, and set the Dimension Sort Order as described in the following table:

**Table 7** Dimension Sort Order Settings

| Dimension Type | Setting |
|---|---|
| Measure | 1 (Default) |
| AllocationType | 2 (Default) |

| Dimension Type | Setting |
| --- | --- |
| POV and Business Dimensions | 3 or higher |
| Attribute Dimensions | Sort as the last dimensions.<br>For example, if you have four attribute dimensions in a sequence of 12 dimensions, the attribute dimensions must be set as 9, 10, 11, and 12. |

4 **Validate the model. See "Validating and Deploying" on page 56.**

# 4 Managing Applications and Dimensions Using Performance Management Architect

**In This Chapter**



## Working with Applications and Dimensions

Profitability and Cost Management uses Performance Management Architect to select dimensions to build the Essbase outline that is used for the profitability model. All dimensions and members are created in Performance Management Architect, and imported into the Profitability and Cost Management application to build the model.

From Performance Management Architect, you can perform the following tasks:

- Create, edit and copy dimensions
- Set up aliases
- Create, view, and delete dimension associations
- Create, view, rename, and delete members
- Edit property values
- Deploy applications to Profitability and Cost Management
- View transaction logs
- Synchronize data between Profitability and Cost Management and other applications, Essbase databases (ASO and BSO), external source (flat files), and interface tables

Refer to the following sections for more information on working with metadata and dimensions using Performance Management Architect:

- "Importing Metadata" on page 39
- "Profitability and Cost Management Dimension and Member Properties" on page 40
- "Deploying Profitability and Cost Management Applications" on page 57

- "Synchronizing Data" on page 59

# Optimizing Dimension Settings for Essbase

A typical Profitability and Cost Management application contains one Measures dimension, one AllocationType dimension, several POV dimensions and a number of business dimensions. Profitability and Cost Management duplicates business dimensions if they are used in more than one stage. This process increases the Sparsity of the Essbase Calculation Cube outline generated by the application, and may have a performance impact when the calculation scripts are run.

Some optimization can be achieved simply by changing the Dimension Storage Type directly for the generated Calculation Cube outline, using Oracle Essbase Administration Services (EAS) console.

**Caution!** Changes to the Dimension Storage Type property should only be performed by a Database Administrator (DBA). For detailed instructions, refer to the *Oracle Essbase Administration Services Developer's Guide*.

The default recommendation to set the Dimension Storage Type property for these dimensions follows:

- Set the Measures and the AllocationType dimensions to Dense
- Set all the business and POV dimensions to Sparse

This default setting results in an Essbase calculation cube with the block size of about 3K, and a large number of potential blocks based on the dimensionality of the Sparse dimensions. The most dense dimension (based on the existence of data) in the largest stage should be set to Dense.

**Note:** The block size should still remain within the limits of the Essbase best practices recommendations. For details, refer to the *Oracle Essbase Administration Services Developer's Guide*.

**Example**

The largest stage is defined by the largest number of potential nodes in the stage.

If a typical Profitability and Cost Management model has its last stage defined as Customers X Products (to calculate Customer x Product profitability), and it is the largest stage, then either the Customer or the Product dimension in this stage can be set to Dense. The associated Measures and AllocationType dimensions can be set to Sparse. The decision on whether to set Customers to Dense or Products to Dense is based on the data density of the governing drivers.

For example, if Sales Volume is the driver that is used to allocate to this stage from various sources in previous stages, and the Customers dimension has more dense data than the Products dimension (few Products sold to more Customers), the Customers dimension can be set to Dense. If more Products are sold to fewer Customers, then Products can be set to Dense. In

either case, the Essbase block size should remain within the limits of the Essbase best practices recommendations.

**Note:** Dimensions with attribute associations cannot be set to Dense. Essbase allows attribute associations to be set as Sparse dimensions only.

# Importing Metadata

The Dimension Library in Performance Management Architect provides a central location for administrators to view, create, and manage dimensions and hierarchies. You can use existing dimensions that are shared with other applications, or create local dimensions that are only for your model.

Performance Management Architect has two types of dimensions:

- Shared dimensions are linked to the Shared Library and inherit all changes that are made to the dimension.
- Local dimensions are copied from the Shared Library to the application. Local dimensions do not inherit any future changes made to the dimension in the Shared Library.

**Caution!** All Profitability and Cost Management properties are local values. If you make changes to this value, it is not automatically inherited if the property exists in other hierarchies. If you modify a property in one hierarchy, you cannot assume that the values will be inherited to other hierarchies.

Before you create applications, you must add dimensions to the Dimension Library. You can create dimensions in several ways, but only in the Dimension Library.

**Caution!** To use the Dimension Library, you must have the Dimension Editor security role. This role permits access to all Dimension Library functionality, such as creating, deleting, and changing dimensions and members, creating import profiles, and running transaction logs. See the *Oracle Hyperion Enterprise Performance Management System Security Administration Guide*.

From the Dimension Library, you can edit these properties:

- Applications
- Dimensions
- Members

**Note:** Standard dimensions for Time, Currency and Country are available in Performance Management Architect for all products.

➤ To access the Dimension Library, select **Navigate**, then **Administer**, then **Dimension Library**.

See *Oracle Hyperion Enterprise Performance Management Architect Administrator's Guide* for detailed instructions on working with dimensions and members.

# Profitability and Cost Management Dimension and Member Properties

The properties for Profitability and Cost Management dimensions and members are displayed in alphabetical order in Table 8, "Profitability and Cost Management Dimension and Member Properties," on page 40, which displays the following information:

- The Property Label, which provides a more readable display name for the property. If applicable, the associated cube type is appended to the name (ASO or BSO). If no cube type is specified, the property applies to both types.
- A Description of each property
- The Property Name, which provides a unique identifier for the property that is used when updating data in the import and batch client.

You can modify any property that presents a drop-down list or data entry text box when you select the property in the Dimension Library.

See *Oracle Hyperion Enterprise Performance Management Architect Administrator's Guide* for detailed instructions on working with dimensions and members.

**Caution!** All Profitability and Cost Management properties are local values. If you modify a property in one hierarchy, you cannot assume that the values will be inherited by other hierarchies.

**Table 8 Profitability and Cost Management Dimension and Member Properties**

| Property Label | Description | Property Name |
|---|---|---|
| Alias | Enter the alias for the selected dimension or member. The alias is the dimension name that displays in a deployed application.<br>● To assign or change the default alias name for the current dimension or member, click the existing alias name and enter the new one. Follow the proper naming rules. All databases have an alias table named Default.<br>● To assign or change an alias name for the current dimension or member, click the existing alias name next to the alias table | Alias |

| Property Label | Description | Property Name |
|---|---|---|
| | and enter the new name. Follow the proper naming rules. | |
| Attributes | Enter the characteristics of a dimension member.<br>For example, Employee dimension members may have attributes of Name, Age, or Address. Product dimension members may have several attributes, such as a size and flavor. | Attributes |
| Attribute Type | Enter the type of attribute for the selected dimension or member:<br>● Text<br>● Boolean<br>● Date<br>● Numeric | AttributeDataType |
| Comment | Enter a comment for the dimension or member. Comments can contain up to 255 characters. By default, this text box displays the current comment, if one exists.<br>You cannot assign a comment to an attribute dimension or member.<br>**Note:** For Unicode-enabled databases, a maximum of 80 characters is allowed. | Comment |
| Consolidation | Select the type of consolidation for the selected dimension member:<br>● + (addition) - Default<br>● - (subtraction)<br>● * (multiplication)<br>● / (division)<br>● % (percent)<br>● ~ (ignore during consolidation)<br>● ^ (never consolidate)<br>● NotUsed<br>**Note:** There are some restrictions with use of consolidation operators in aggregate storage outlines. Refer to the *Oracle Essbase Database Administrator's Guide* for additional information. | Consolidation |
| Data Storage(ASO) | Select the ASO storage options for dimensions and members: | ASODimensionDataStorage |

| Property Label | Description | Property Name |
|---|---|---|
| | • StoreData – Data is stored with the dimension.<br>• ShareData — Data associated with this member can be shared. The ShareData property applies to the member Only. The Dimension Root Member cannot be shared.<br>• NeverShare – Data associated with this dimension cannot be shared, even if there is an implied share relationship, such as with a parent with one child. In this case, the data is duplicated in the parent and child.<br>**Note:** This option does not apply to stored hierarchies in aggregate storage outlines.<br>• LabelOnly – No data is associated with this dimension. | |
| Data Storage(BSO) | Select the BSO storage options for dimensions and members:<br>• StoreData – Data is stored with the dimension.<br>• ShareData — Data associated with this member can be shared. The ShareData property applies to the member Only. The Dimension Root Member cannot be shared.<br>• NeverShare – Data associated with this dimension cannot be shared, even if there is an implied share relationship, such as with a parent with one child. In this case, the data is duplicated in the parent and child.<br>• LabelOnly – No data is associated with this dimension.<br>• DynamicCalcAndStore – Data associated with this dimension is not calculated until requested by a user. Once calculated, the data is stored.<br>• DynamicCalc – Data associated with this dimension is not calculated until requested by a user. The data is not stored but is discarded after the request is completed. | DataStorage |

| Property Label | Description | Property Name |
|---|---|---|
| Dimension Formula(ASO) | Double-click the cell and click the Selector button to open the Memo Editor. Enter the formula, then click OK.<br>Use the appropriate ASO MDX syntax. | ASODimensionFormula |
| Dimension Formula(BSO) | Double-click the cell and click the Selector button to open the Memo Editor. Enter the formula, then click OK.<br>Use the appropriate BSO CALCULATOR syntax. | BSODimensionFormula |
| Dimension Solve Order | Enter the numeric value in the solution sequence for selected dimensions. For example, if this dimension is to be solved second, enter “2.” | DimensionSolveOrder |
| Dimension Sort Order | Enter the numeric value in the sequence to set the order of dimensions in the Essbase outline that is generated by Profitability and Cost Management.<br>For example, if this dimension is to be the second in the Essbase outline, enter “2.”<br>Dimension Sort Order must be set for every dimension in the model, except Alias and UDA dimensions. The dimension sort order must be sequential.<br>Set the sort order as follows:<br>● Measures dimension is set to 1, by default.<br>● AllocationType dimension is set to 2, by default.<br>● POV and business dimensions are set to 3 or higher.<br>● Attribute dimensions are sorted as the last dimensions.<br>For example, if you have four attribute dimensions in a sequence of 12 dimensions, the attribute dimensions must be set as 9, 10, 11, and 12. | DimensionSortOrder |
| Dimension Storage Type | Select the type of storage required for the dimension:<br>● Dense<br>● Sparse | DimensionStorageType |

| Property Label | Description | Property Name |
|---|---|---|
| | **Note:** When you define the value for this property at the parent level, all children automatically inherit the value. | |
| Hierarchy Type<br>(Dimensions Only) | Set the type of hierarchy for the dimension:<br>● Enabled<br>● Disabled<br>● Stored<br>● Dynamic | DimensionHierarchyType |
| Hierarchy Type<br>(Members Only) | Set the type of hierarchy for the member:<br>● Stored (For first Gen2 child)<br>● Dynamic (For Second Gen2 child onwards. The Second Gen2 member can host the alternate hierarchies)<br>● None (Do Not Use)<br>**Note:** The last Gen2 child must always be '`NoMember`'. | HierarchyType |
| Member Formula(ASO) | Double-click the cell and click the Selector button to open the Memo Editor. Enter the formula, then click OK.<br>Use the appropriate ASO MDX syntax. | ASOMemberFormula |
| Member Formula(BSO) | Double-click the cell and click the Selector button to open the Memo Editor. Enter the formula, then click OK.<br>Use the appropriate BSO (BSO CALCULATOR syntax. | BSOMemberFormula |
| Member Solve Order<br>(DimensionsOnly) | Enter the numeric value in the solution sequence for selected member. For example, if this member is to be solved second, enter "2."<br>This property applies to ASO databases only. | MemberSolveOrder |
| Member Solve Order<br>(Members Only) | Enter the numeric value in the solution sequence for selected member. For example, if this member is to be solved second, enter "2."<br>This property applies to ASO databases only. | MemberSolveOrder |

| Property Label | Description | Property Name |
|---|---|---|
| POV Dimension | Click the checkbox to set this dimension as a POV dimension for the selected model. | IsPOVDimension |
| POV Display Order | If there are multiple POV dimensions, enter the numeric value (such as 1, 2, 3, and so on) to set the display order for each POV dimension. | POVDisplayOrder |
| Primary Level Weighting | Primary level weighting restricts the levels which can be selected by the view selection engine during aggregation operations. These settings only apply to Essbase (ASO) applications.<br>Select one of the following options:<br>● Default – The view selection engine is free to decide which levels to pick. This is the initial value for new hierarchies.<br>● AllLevels – The view selection engine considers all levels of the hierarchy for aggregation.<br>● NoAggregation – The view selection engine cannot select any levels for aggregation.<br>● TopLevelOnly – The view selection engine considers only the highest level of the dimension for aggregation.<br>● BottomaTop – The view selection engine considers only the highest level and the lowest level of the dimension for aggregation.<br>● BottomLevelOnly – The view selection engine considers only the bottom level of the dimension for aggregation. | PrimaryLevelWeighting |
| Two Pass Calc(BSO)<br>(Dimensions Only) | Select the checkbox to enable two pass calculations on dimensions. | TwoPassCalc(BSO) |
| Two Pass Calc(BSO)<br>(Members Only) | Select the checkbox to enable two pass calculations on members. | TwoPassCalc(BSO) |
| UDA | Enter the UDA name for the selected member. A UDA dimension must exist in Performance Management Architect, and be associated to the base to which the UDA is being added. See “Viewing Dimension Associations”. | UDA |

| Property Label | Description | Property Name |
|---|---|---|
| | You can enter up to 256 alphanumeric characters. Multiple UDAs must be comma separated. For example: Small Market,New Market.<br>**Note:** For Unicode-enabled databases, a maximum of 80 characters is allowed. | |

# Creating Profitability Applications

You can create Profitability applications in Performance Management Architect using one of two methods:

- Create the Profitability and Cost Management application using the application wizard to automatically create the application and dimensions.
- Create a blank application, and manually select and name the dimensions.

**Caution!** Any business dimensions to be included in the application, for example, Generic, Account, Entity, Time, or Country, must be added to the Dimension Library before creating the application; otherwise, the dimensions will not be available for the Application Wizard to select.

The following sections provide step-by-step procedures for creating Profitability applications:


- "Selecting the Application Type" on page 46
- "Selecting Dimensions" on page 47
- "Modifying the Application Settings" on page 52
- "Validating and Deploying" on page 56


## Selecting the Application Type

➤ To create Profitability and Cost Management applications:

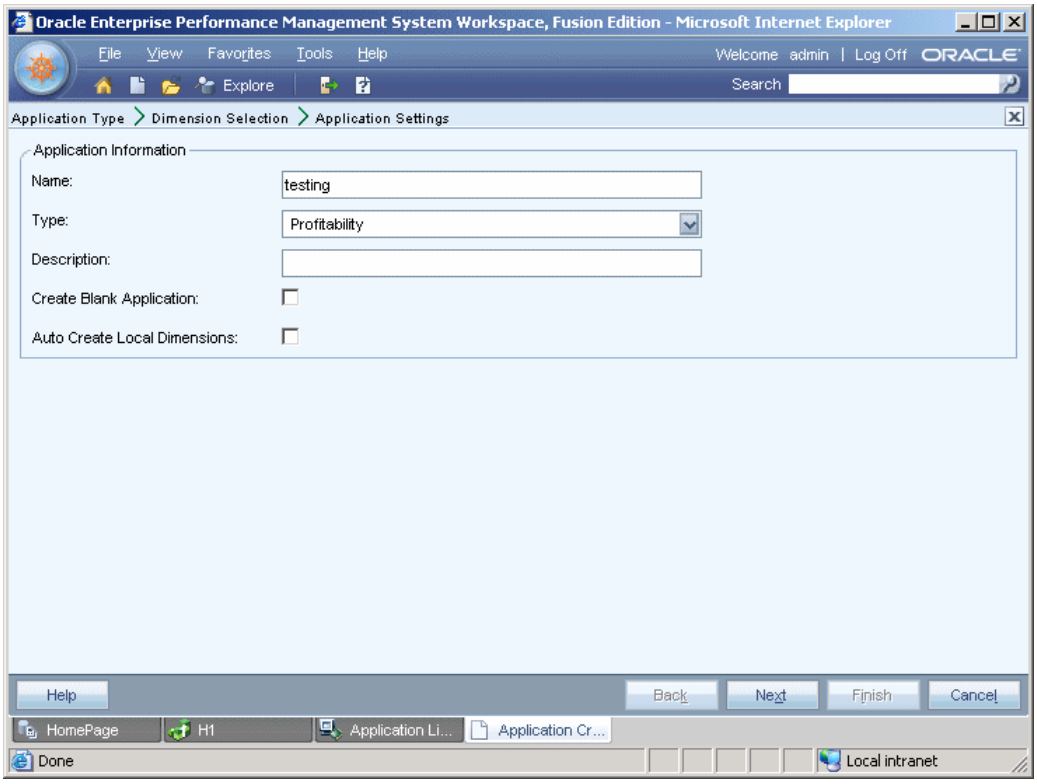
1 **Populate the new shared library in Performance Management Architect using a flat file import or a Performance Management Architect interface table import.**

2 **From EPM Workspace, select Navigate, then Administer, and then Application Library.**

The Application Library is displayed.

3 **Select File, then New, and then Application**

The first screen of the wizard is displayed - Application Type.




4 Under **Name**, enter the application name.

Names must be 7 characters or less, and must not contain any special characters, including "&" (ampersands).

5 Under **Type**, select **Profitability**.

6 Under **Description**, enter a description.

7 **Optional**: To manually add dimensions to the blank application, select **Create Blank Application**, and then click **Finish**. See "Adding Dimensions Manually " on page 51.

8 **Optional**: To automatically create all required dimensions, select **Auto Create Local Dimensions**, and then click **Next**. See "Selecting Dimensions" on page 47.

Selecting Auto Create Local Dimensions automatically creates new dimensions for all dimensions that are required in the application. The dimension name for each new dimension is identical to the dimension type with (New) in parentheses.

The second screen of the wizard is displayed - Dimension Selection.

9 Click **Next**. See "Selecting Dimensions" on page 47.

## Selecting Dimensions

After creating the application, you must select the dimensions that are to be included in the application.

The following dimensions are required for Profitability and Cost Management applications:

- Measure dimension contains the dimensions and members required to build, validate and calculate a model, including driver measures, reporting measures, and allocation measures. Users may add any user-defined driver measures (or sub-hierarchies) to the hierarchy under the member 'UserDefinedDriverMeasures'.
- AllocationType dimension is used to store direct allocations and genealogy allocations.
- POV dimensions provide a specific view of your model information for a selected time period, such as a year, status and scenario. At least one Point of View (POV) dimension must be defined by the user. You can also create a POV Version.
- At least one Business Dimension must be defined by the user. Business, or user-defined, dimensions contain members that store information that is specifically related to the requirements of your business or organization, such as product types, sales regions, manufacturing processes, general ledger, payroll, departments, and so on. This dimension type does not apply to aggregate storage outlines.

  Business dimensions may include some or all of the following dimension types, and may apply to one or more stages or models:

  - Generic
  - Account
  - Entity
  - Time
  - Country

  **Note:** Although these business dimensions can be included as part of a Profitability and Cost Management application, when the Essbase outlines are deployed, they are created as basic or generic dimensions, with no type.
- Attribute dimensions are a special type of dimension that are associated with a business dimension. Attributes describe characteristics of data, such as the size and color of products.
- Alias is optional, and only required if you want to use aliases in your model.

**Note:** Model stages are not dimensions, and are therefore not available in Performance Management Architect. Stages are added to a model in Profitability and Cost Management, and are used to organize dimensions into logical steps or stages.

After creating the application, select the dimensions to be added to the application using the appropriate procedure:

- If you are using the wizard, see "Adding Dimensions Using the Wizard" on page 48.
- If you created a blank application, see "Adding Dimensions Manually " on page 51.

## Adding Dimensions Using the Wizard

When you use the application wizard, all required dimensions are automatically displayed. If there is an exact match, it automatically populates the dimension column for the dimension

type. The required dimension types for Profitability and Cost Management are automatically categorized and displayed with a shaded heading:

- Measure Dimension
- AllocationType Dimension
- POV Dimension
- Alias Dimension
- Business Dimensions
- Attribute Dimensions

If you selected Auto Create Local Dimensions when selecting the application type, new local dimensions are created for each required dimension. The name of each new dimension is the same as the dimension type, with (New) in parentheses. For example, Account (New).

Any business dimensions you want to include in the application, for example, Account, Entity, Time, or Country, must be added to the Dimension Library before creating the application; otherwise, the dimensions will not be available for the Application Wizard to select.

➤ To define dimensions:

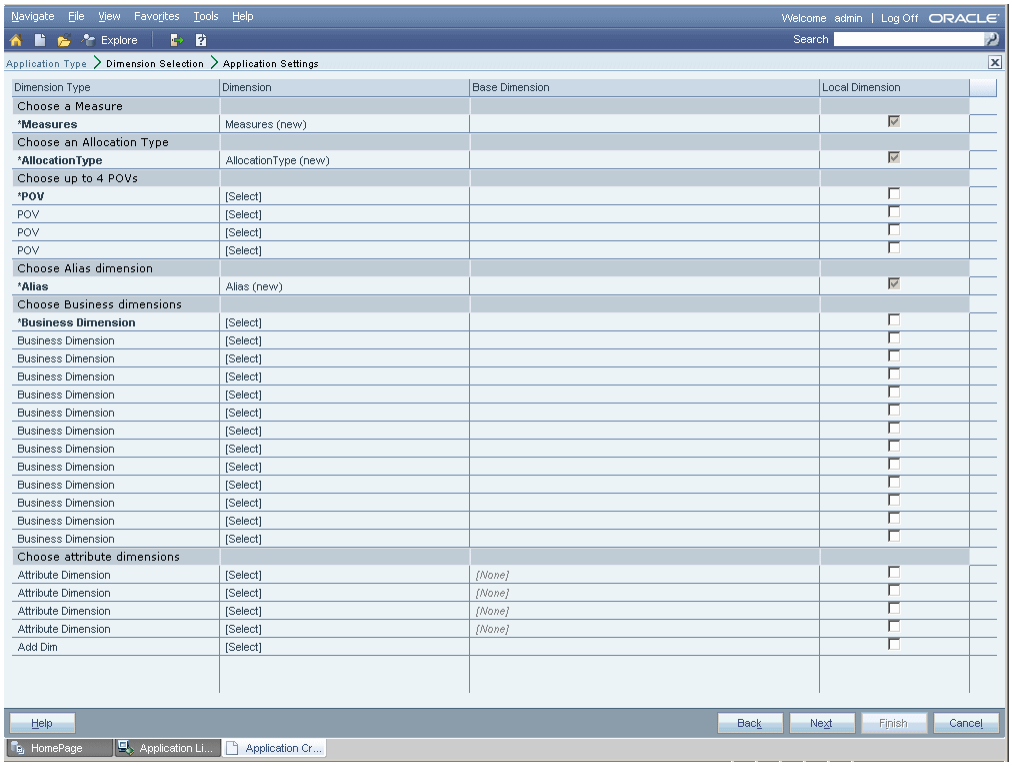
1 On the Dimension Selection tab (Step 2 of the Application Wizard), under **Dimension Type**, review the required dimension types.

2 For each existing dimension to be added for the application, under the **Dimension** column, click **[Select]** to display the drop-down list of available dimensions for that dimension type.

Only the dimensions for the associated dimension type are displayed. For example, if you are mapping a dimension to the Measures dimension type, only measure dimensions display in the list.

3 Select a dimension from the drop-down list.

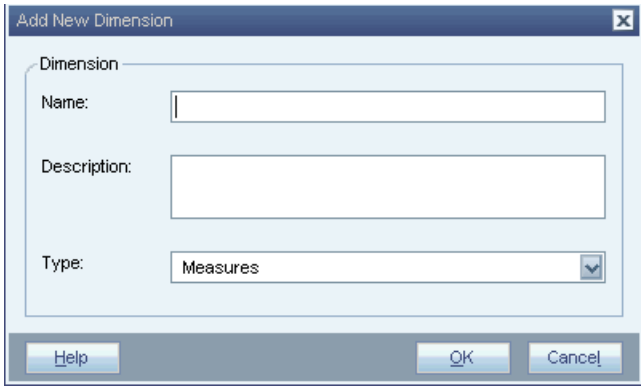
4 **Optional**: Create a new dimension. To create a new dimension:




a. In the **Dimension** column, click [**Select**] to display the drop-down list of available dimensions for that dimension type.

b. From the drop-down list, select [**Create New Dimension**].

The Add New Dimension dialog box is displayed.




c. Enter the **Name** and **Description** of the new dimension. The **Type** is automatically selected, based on the dimension type.

d. Click **OK**.

5 **Repeat step 3 and step 4 to select or create new dimensions.**

**Note:** All dimensions must be mapped to a dimension or [None] before you can move to the next screen.

6 **Optional:** To designate the dimension as a local dimension, select the **Local Dimension** check box at the end of each row.

For information on shared and local dimensions, see "Selecting Dimensions" on page 47.

7 Click **Next**.

The Application Settings screen is displayed.

## Adding Dimensions Manually

When you create blank applications, you use the Dimension Library to select dimensions.

The Dimension Library has three panes:

- **Shared Library** - Displays on the left and displays all dimensions currently available in the Shared Library.
- **Application** - Appears in the middle and when expanded, lists the names of all dimensions in the current application.
- **Property Grid** - Displays on the right and displays the properties of the selected dimension or member.

For detailed information on using the Dimension Library, see the *Oracle Hyperion Enterprise Performance Management Architect Administrator's Guide*

➤ To add dimensions to blank applications:

1 In the Dimension Library, drag the dimensions from the Shared Library to the application.

The Copy Dimension dialog box is displayed.

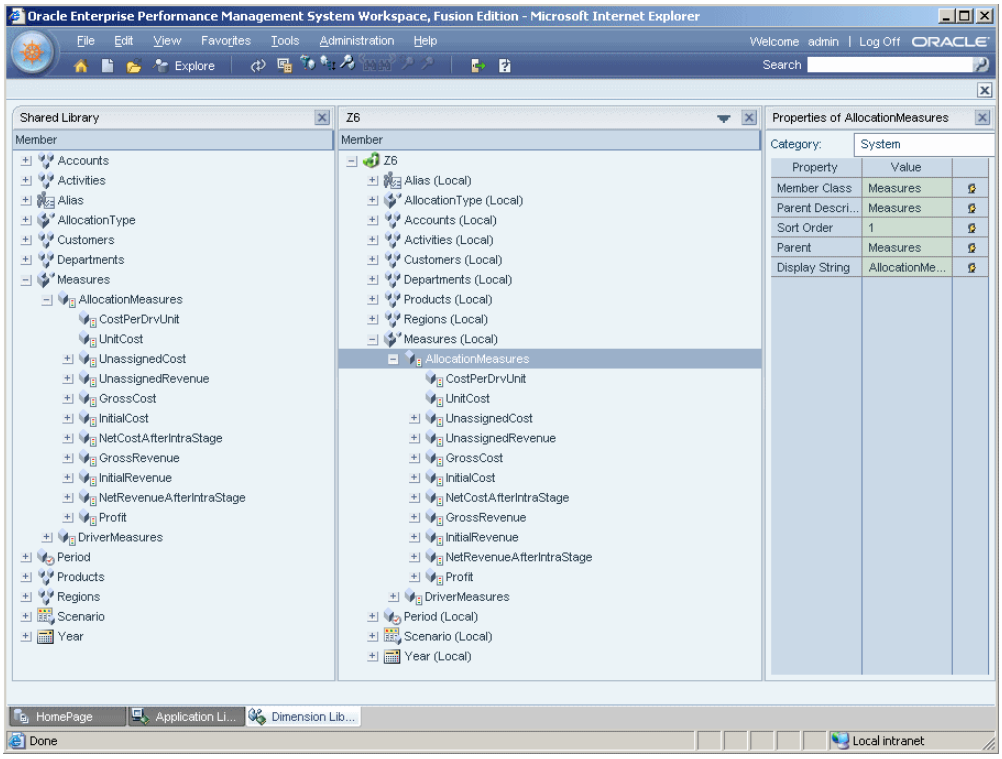
**Tip:** To create new local dimensions, right-click the application name in the Application View, and select Create Dimension. The Add New Dimension dialog box is displayed.

2 Select Local Dimension or Shared Dimension.

**Tip:** To add only members to an application, in the Shared Library, right-click the dimension and select Add to App View.

3 Add all required dimensions to the application. (Measures, AllocationType, POV, and so on.)

The expanded application lists all dimensions.




**Tip:** If you add a member to the Shared Library and include the dimension in the application, click the Refresh button  , and the new member will display in the application.

4 Set the **Consolidation Property Value** for **NoMember** to **~** (Ignore).

5 Reorder the dimensions so that `NoMember` is moved to the last Gen2 position. See "Reordering Children Members" on page 55.

6 Modify the application settings, as required. See "Modifying the Application Settings" on page 52.

## Modifying the Application Settings

The next step in creating an application is to modify application properties, associations, and other application settings. If you are creating a blank application and using the Dimension Library to modify the application settings, you have similar options, except that you are using the Dimension Library instead of the wizard to make the modifications.

**Note:** Dimension associations are required for shared hierarchies, and may not be required in all models.

See the following procedures to modify application settings:

- Use the Dimension Library for blank applications. See "Modifying Application Settings Using the Dimension Library" on page 53.

- Use the Application Wizard for auto-generated applications. See "Modifying Applications Settings Using the Application Wizard" on page 53.

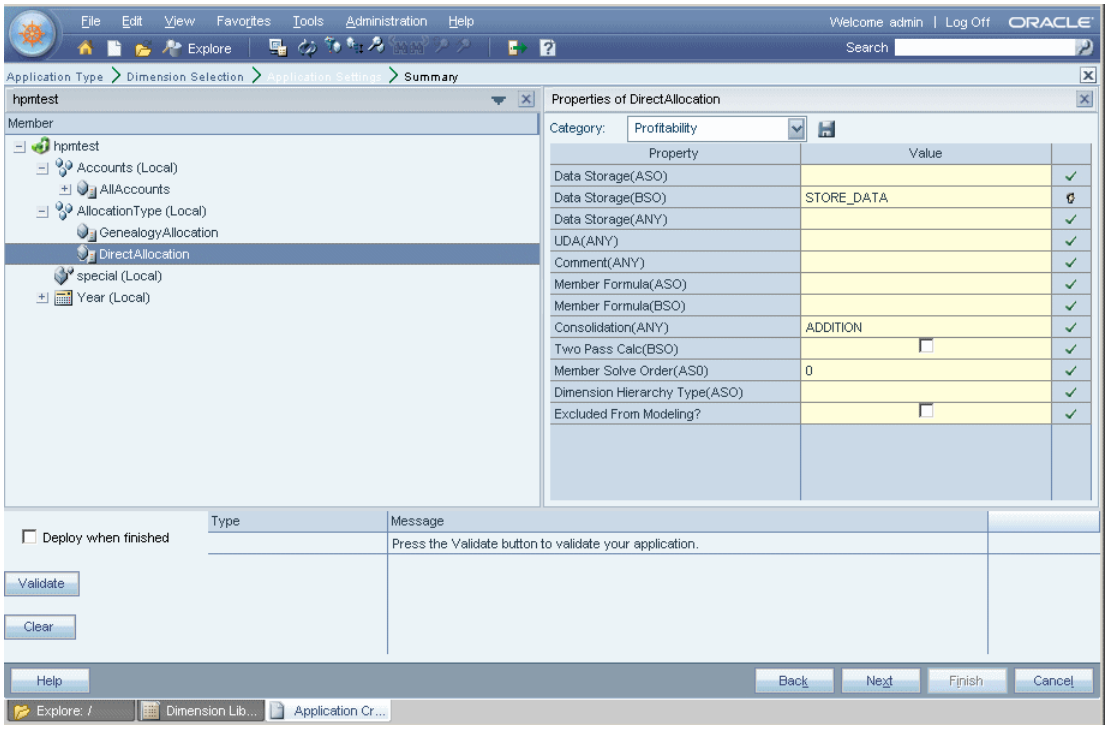
## Modifying Application Settings Using the Dimension Library

In the Dimension Library, the Shared Library, Application and Property Grid are displayed. You can use the right-click menu to modify the application dimensions and perform these actions

- For a Profitability and Cost Management model, you must perform the following tasks:
  - Set the Properties for POV Dimensions, and the POV Display Order for multiple POV dimensions, if required. See "Setting the Properties for POV Dimensions" on page 55.
  - Set the Dimension Sort Order for all model dimensions. See "Setting the Dimension Sort Order" on page 35.
  - Reorder the `NoMember` to display this member as the last generation 2 member on the list. See "Reordering Children Members" on page 55
- Modify dimensions, as required, using the procedures provided in the *Oracle Hyperion Enterprise Performance Management Architect Administrator's Guide*:
  - Set the dimension alias.
  - View application membership.
  - View dimension associations.
  - Synchronize dimensions.
  - Create members.
  - Insert shared members.
  - Manage orphan members.

## Modifying Applications Settings Using the Application Wizard

If you are creating the application using the wizard, the application dimensions and members display on the left, the Property Grid displays on the right, and the validation and deployment options display at the bottom.




You can use the right-click menu to modify the application dimensions. For detailed instructions to perform these tasks, see the *Oracle Hyperion Enterprise Performance Management Architect Administrator's Guide*:

- For a Profitability and Cost Management model, you must perform the following tasks:
  - Set the Properties for POV Dimensions, and the POV Display Order for multiple POV dimensions, if required. See "Setting the Properties for POV Dimensions" on page 55.
  - Set the Dimension Sort Order for all model dimensions. See "Setting the Dimension Sort Order" on page 35.
  - Reorder the `NoMember` to display this member as the last generation 2 member on the list. See "Reordering Children Members" on page 55
- Modify dimensions, as required, using the procedures provided in the *Oracle Hyperion Enterprise Performance Management Architect Administrator's Guide*:
  - Set the dimension alias.
  - View application membership.
  - View dimension associations.
  - Synchronize dimensions.
  - Create members.
  - Insert shared members.
  - Manage orphan members.

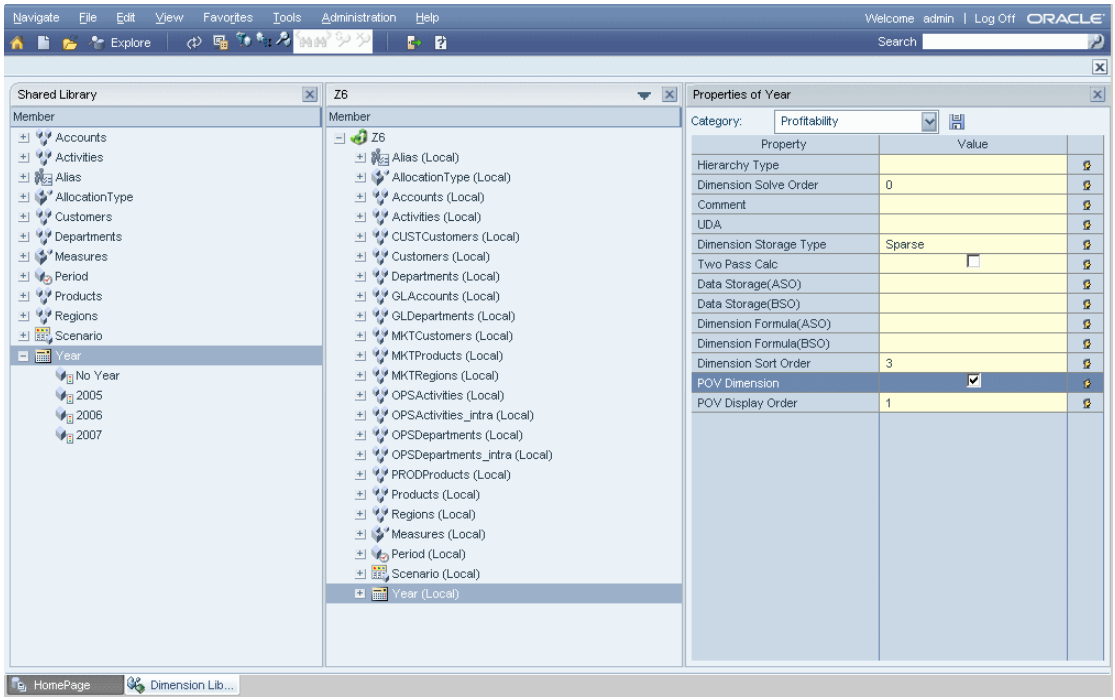
## Setting the Properties for POV Dimensions

At least one POV, or Point of View, dimension must be set for every Profitability and Cost Management model. The POV dimension may be set to any custom value, but usually denotes time periods, such as Year, Period, or Scenario. You can also create a POV Version.

If more than one POV dimension is designated, a POV Display Order must also be set to sequence these dimensions for calculation. The POV Display Order that is set in Performance Management Architect is automatically picked up in Profitability and Cost Management.

➤ To set POV dimension properties:

1. **Select the first POV dimension, such as Year.**
2. **In the Property Grid under Category, select Profitability.**
3. **Select POV Dimension.**




4. **Optional:** Repeat step 1 to step 3 to select other POV dimensions, such as Period or Scenario.
5. **Optional:** If more than one POV Dimension was selected, set the **POV Display Order** for each POV Dimension.

   Double-click the POV Display Order cell and type the sequence number for the selected POV dimension. For example, if Year, Period and Scenario are set as POV Dimensions, set the POV Display Order for Year to 1, Period to 2, and Scenario as 3 as the display order.

## Reordering Children Members

For all business dimensions in the model, you must reorder the list of members to display `NoMember` as the last gen 2 member; otherwise, validation of the model will fail.

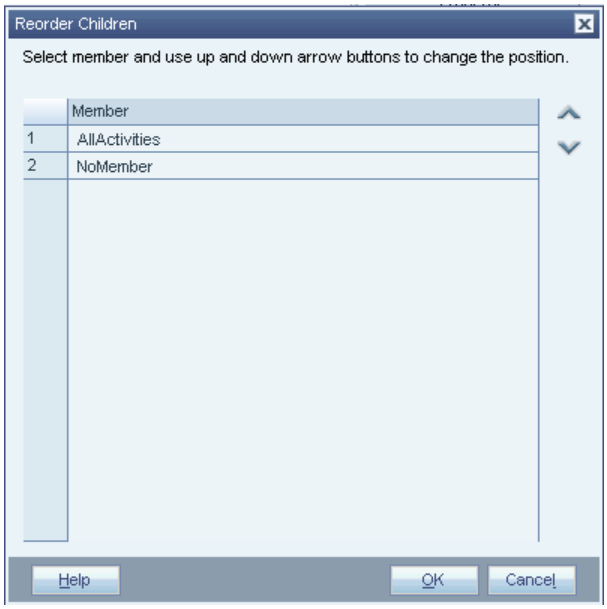
You can also reorder members to suit your particular requirements as long as `NoMember` remains in the last position.

**Note:** This step is not required for system dimensions, such as Alias, AllocationType, Measures, Periods, Scenarios or Years.

➤ To reorder children members:

1. From EPM Workspace, select **Navigate** , then **Administer**, and then **Dimension Library** to display the Shared Library.
2. In the Shared Library, right-click the dimension that you want to reorder.

   The Reorder Children dialog box is displayed.




3. Select the member to be moved, and use the up and down arrows to change the order of the children.

   **Note:** Ensure the `NoMember` dimension is moved to the end of the list as the last gen 2 member; otherwise, the deployment will fail.

4. Repeat step 3 to order all members.
5. Click **OK**.

## Validating and Deploying

After the application settings have been modified, validate and deploy the application, as follows:

- If you are using the application wizard, validate the application and correct any errors prior to finalizing the creation of the application. If required, you can bypass the validation and deployment options in the wizard and later use the deployment option in the Application Library.

- If you are creating a blank application, validate and deploy the application using options in the Application Library, as outlined in the *Oracle Hyperion Enterprise Performance Management Administrator's Guide*.

### Profitability and Cost Management Validations

For Profitability and Cost Management, the following conditions are validated:

- The name of the application must be 7 characters or less, and contain no special characters.
- At least one dimension has been set to POV type.
- Up to 4 dimensions may be marked as POV dimensions
- The application must contain one Measures dimension
- The application must contain one AllocationType dimension
- There is only one dimension of type "Account" is allowed
- There is only one dimension of type "Entity" is allowed
- Dimension Sort Order has been set for the model. See "Dimension Sort Order " on page 34
- Ensure that duplicate members do not exist in the same dimension.
- `NoMember` must be set as the last Generation 2 member for all business dimensions, and must be set to Ignore (~) in the Property Grid.

  **Note:** This requirement does not apply to Alias, AllocationType, Measures, Periods, Scenarios or Years.

➤ To validate and deploy an application using the wizard:

1. **On the Application Settings screen of the Application wizard, click Validate.**

   All errors display in the grid, showing the error types and error message.

2. **Optional: To deploy the application after validation, select Deploy when finished.**
3. **If validation errors occur, correct any problems before deploying the application.**
4. **Click Finish.**

   If you selected "Deploy when finished," the Deploy dialog box is displayed.

   **Note:** If you change a shared dimension, all applications that use that shared dimension are affected. For changes to take effect, the application must be redeployed.

5. **Deploy the application. See "Deploying Profitability and Cost Management Applications" on page 57.**

## Deploying Profitability and Cost Management Applications

After creating a Profitability and Cost Management application in Performance Management Architect, the application must be validated and deployed to Profitability and Cost Management.

**Note:** For Level-0 members, you can add, rename, reparent or delete individual members, multiple members, or combinations of members during redeployment of an application; however these actions are not currently available for non-Level-0 members.

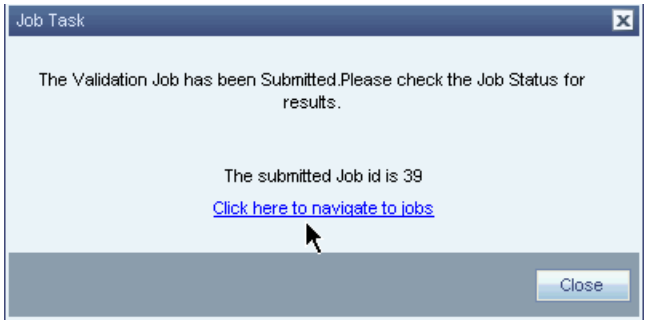
For detailed instructions on working with Performance Management Architect to create and deploy Profitability and Cost Management applications, see the *Oracle Hyperion Enterprise Performance Management Architect Administrator's Guide*.

➤ To deploy a Profitability and Cost Management application in Performance Management Architect:

1. From EPM Workspace, select **Navigate**, then **Administer**, and then **Application Library** to display the Application Library.
2. From the Application Library, select the application to be deployed.
3. Right-click the application name, and select **Validate** to validate the newly created Profitability and Cost Management application in Performance Management Architect.

   For Profitability and Cost Management validation conditions, see "Validating and Deploying" on page 56.

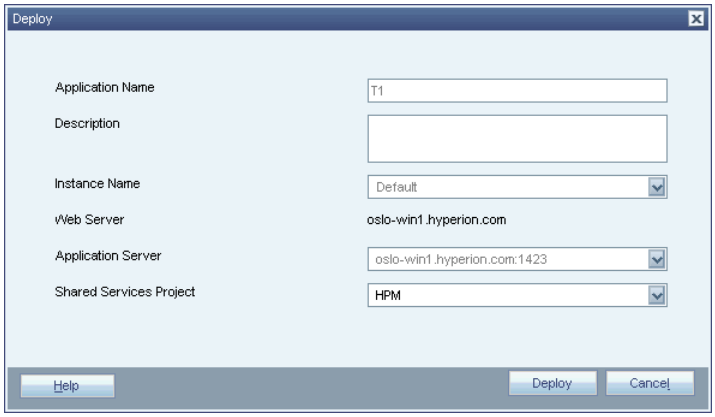
   When the validation is complete, the Job Console is displayed, showing the associated job number.




4. Click the link for the associated job to view the Job Console.
5. Review the Job Console Summary, and if there are errors, click the validation log link under **Attachments** to see a complete listing of all validation errors or messages.
6. Fix any errors listed in the Validation Log for the job number.
7. Repeat step 3 to step 6 until the validation succeeds with no errors.

   The application is ready to be deployed.
8. In the Application Library, right-click the application name, and select **Deploy** from the drop-down list.

   The Deploy dialog box is displayed.




9 **Provide the following information for the deployment:**

- Enter the name of the application being deployed
- **Optional:** Enter a Description of the application.
- Under **Instance Name**, select the name of the Profitability and Cost Management installation to which you want to deploy the application.

  A Profitability and Cost Management application may be installed on a single machine, or on more than one machine as part of a cluster. Each installation, or instance, is displayed on the Instance Name drop-down list. Default is the name of the instance used by the Configurator Tool for the first installation.
- Select the **Application Server.**
- Under **Shared Services Project**, select the Profitability and Cost Management application group.

**Note:** The Profitability and Cost Management group is not displayed the first time the deployment is selected. You must create the Profitability and Cost Management application group or project in the Shared Services Console.

10 **Click Deploy.**

The Job Task window is displayed, to confirm the Validation Job has been submitted, and display the Job ID.

11 **On the Job Task window, click the link to display the job status.**

When the job is complete, a message is displayed under Detail to indicate the new application has been created and deployed to Profitability and Cost Management. The new application is available for selection.

# Synchronizing Data

Data synchronization enables you to synchronize and map data between Oracle applications, interface tables, and external files.

Using the Data Synchronization module in Performance Management Architect, you can synchronize data between Oracle Hyperion Financial Management, Fusion Edition, Planning,

Profitability and Cost Management, Essbase (ASO and BSO) as destinations, and the following sources:

- Oracle Hyperion Financial Management, Fusion Edition
- Oracle Hyperion Planning, Fusion Edition
- Profitability and Cost Management
- Essbase (ASO and BSO)
- External source (flat file)
- Interface table

Data synchronization currently supports data transfer from BSO cubes only.

See *Oracle Hyperion Enterprise Performance Management Architect Batch Client User's Guide* for detailed instructions on synchronizing data.

# 5 Importing Data in Profitability and Cost Management

**In This Chapter**



You can enter data directly into Profitability and Cost Management; however, the data entry may be very time-consuming. To facilitate the population of the application you can import data directly into Profitability and Cost Management using a set of import staging tables and import configurations.

Model data is imported from several sources:

- Model metadata and dimensions are imported from the Performance Management Architect Dimension Library through EPM Workspace. See the chapter on importing metadata in the *Oracle Hyperion Enterprise Performance Management Architect Administrator's Guide*.
- Model definition data is imported into Profitability and Cost Management. The data may be created in another application, such as Excel, and imported using the staging tables. See Appendix A, "Staging Tables".
- Model and end-user data can be imported and exported from Essbase. See the *Oracle Essbase Database Administrator's Guide*.
- Model data can be imported using Oracle Hyperion Enterprise Performance Management System Lifecycle Management. See the *Oracle Hyperion Enterprise Performance Management System Lifecycle Management Guide*.

**Caution!** Oracle recommends that, before importing data, you create a backup directory of your databases in EPM Workspace and Essbase. See Chapter 7, "Backing Up Profitability and Cost Management Components".

You need to create an import configuration to specify which tables and data are to be imported. The configuration may be saved, and used multiple times to import the same set of data.

For detailed instructions, refer to these sections:



# About Staging Tables

When data is imported into Profitability and Cost Management, users create the staging tables that provide the pre-defined structure to manage the import. The tables are created using a relational database, such as Oracle or SQL Server, to organize the data into a format that can be easily matched to the application.

You can create all tables simultaneously, or create only the tables that you want to import. You must populate at least one of the following data groups:

- Stages
- POV States
- Driver
- Driver Selection
- Assignment
- Assignment Rule Selection

The staging tables are created by the Profitability and Cost Management administrator (*admin*), using the format specified in Appendix A, "Staging Tables".

Tables can be created manually, or by running the SQL script,`create_staging.sql`. This script is used for both SQL and Oracle. After installation, the SQL script is included in the application folder. By default, the location is *Hyperion_Home*`\products\Profitability\database\Common`.

To import data, you must have the appropriate user role and security authorization. See the *Oracle Hyperion Enterprise Performance Management System Security Administration Guide*.

# Creating Import Configurations

Data can be imported only if all of the following are available:

- A source database and its associated staging tables and data
- A target application to receive the data
- An import configuration to define which tables and data are to be imported. The import configuration can be used multiple times.

To streamline importing, consider creating separate configurations for different sets of information. By using smaller import groups, you can reduce import times, and avoid repetitive updates of static information. For example, you might group model elements for an import configuration, as follows:

- Infrequently changed: POV, and stages
- Frequently changed: Drivers, driver selections and assignments.

After a configuration is created, run the configuration to import data into the application.

➤ To create import configurations:

1 **Select a database to be used as the source for the import.**

You can use an existing database that has been formatted for the import, or create a new, blank database.

There are no naming restrictions for the database.

2 **Run the `create_staging.sql` script against the source database to automatically create the staging tables. The tables are generated using the schema provided in Appendix A, "Staging Tables".**

3 **Load the model data into the source staging tables.For detailed instructions, see Importing Data in the *Oracle Hyperion Profitability and Cost Management User's Guide*.**

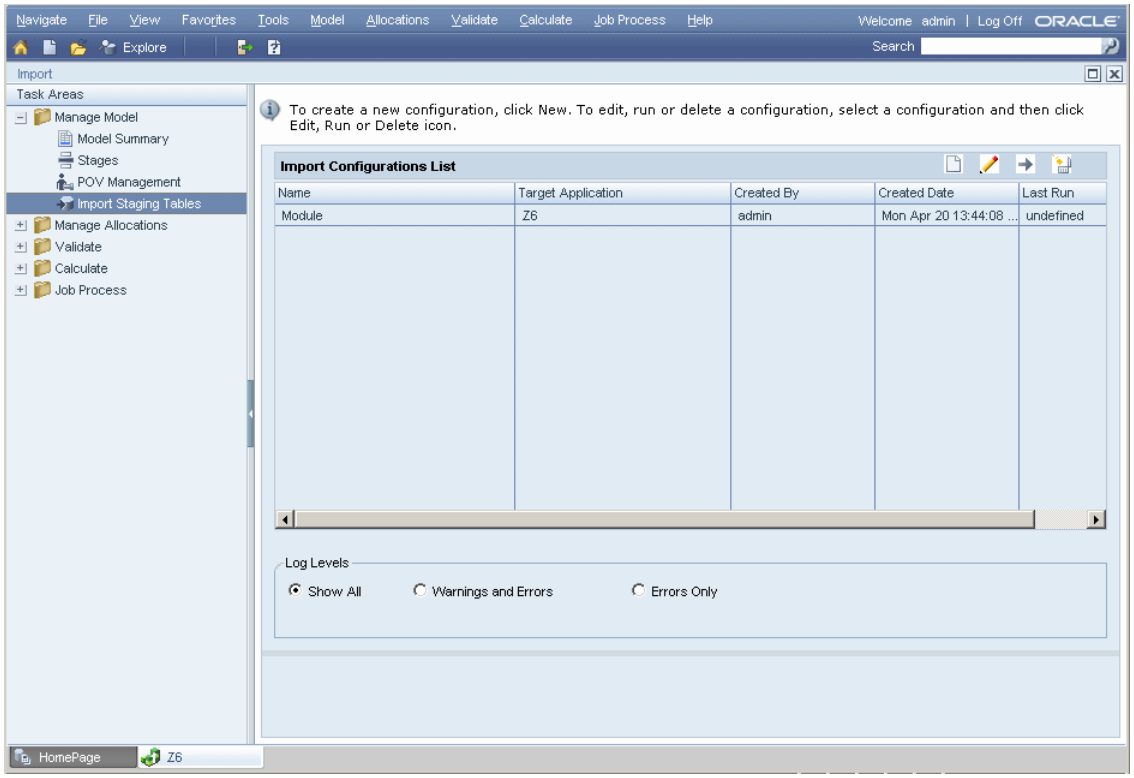
4 **Review the entries in the staging tables for any obvious issues, and remove any null rows from the source database.**

5 **Verify that the source database is accessible.**

6 **In EPM Workspace, select Navigate, then Applications, then Profitability, and then the model name to access the Profitability and Cost Management model.**

7 **From Task Areas, select Manage Model, then Import Staging Tables.**

The Import page is displayed.

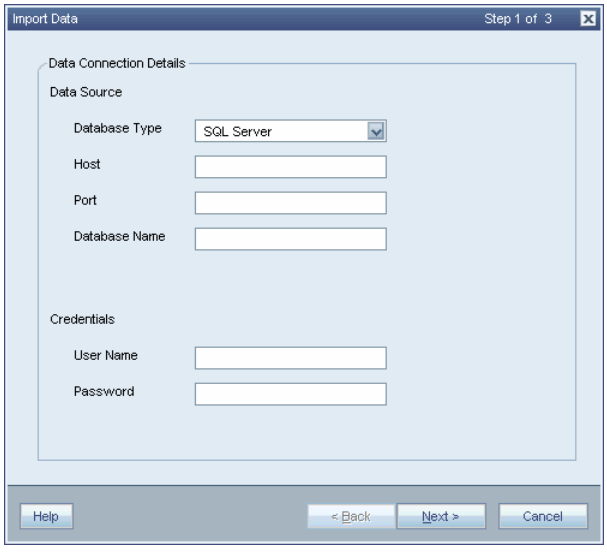



8 Under **Log Levels**, select the level of error messages you want to record for the import process:

- Show All — Display all warnings, errors and information messages.
- Warnings and Errors — Display warning and error messages only.
- Errors Only — Display only messages for errors encountered during the operation.

9 From **Import**, click the Add button to create a new import configuration.

Step 1 of the Import Data dialog box is displayed.



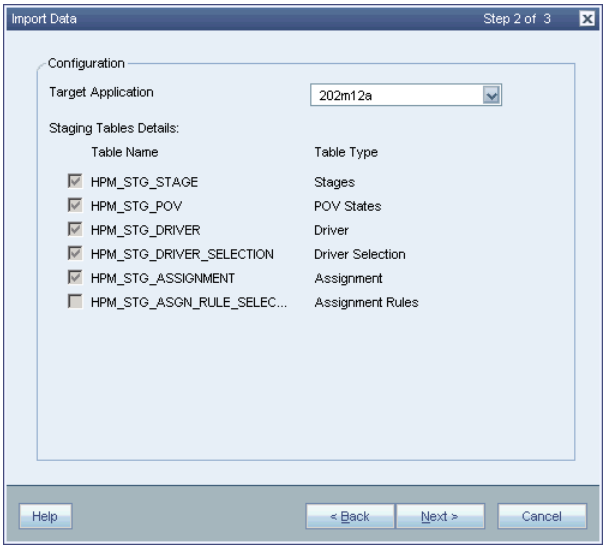

10 Under **Data Source**, enter the access details for the server on which the source staging tables reside:

- From **Database Type**, select the type of database being used for the source database.
- Under **Host**, enter the path to the location of the source database that contains the source staging tables.
- Under **Port**, enter the name of the port associated with the source database.
- Under **Database Name**, enter the name of the source database.

11 Under **Credentials**, enter the **User Name** and **Password** for the authorized user of the source database, if required.

12 Click **Next**.

Step 2 of the Import Data dialog box is displayed.




13 Under **Configuration**, select the **Target Application**.

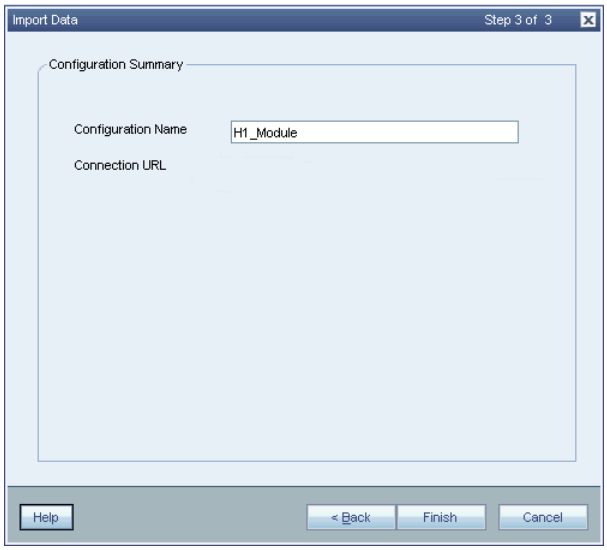
The target application is the Profitability and Cost Management application into which the data is being imported.

14 Under **Staging Tables Details**, select one or more staging tables to be imported.

The name of existing staging table is listed under Table Name, and the type of data in each staging table is shown under Table Type.

15 Click **Next**.

Step 3 of the Import Data dialog box is displayed.




16 Under **Configuration Summary**, enter a **Configuration Name** to store this import configuration.

The database name associated with this configuration is displayed below the configuration name. The configuration name cannot contain more than 80 characters.

17 Click **Finish**.

The import configuration is added to the Import Configurations List, and can be reused to import the same set of data many times.

18 **Optional**: Select the import configuration, and click the Run button to run the import configuration. See "Running Import Configurations " on page 67.

# Modifying Import Configurations

When updating an import configuration, only the target application can be changed. To select a different set of staging tables for the configuration, you must create a new import configuration.

➤ To modify import configurations:

1 From **Task Areas**, select **Manage Model**, then **Import Staging Tables**.

The Import page is displayed.

2 From the **Import Configurations List**, select an import configuration.

3 Click the Modify button .

The Import Data dialog box is displayed.

4 Review the connection information and credentials for the selected import configuration, and then click **Next**.

5 **Optional:** Under **Target Application**, select a target application for this import configuration.

**Note:** You cannot change the staging tables that are selected. To select a different set of staging tables to be imported, you must create a new import configuration.

6 Click **Next**.

7 **Optional:** Under **Configuration Summary**, enter a **Configuration Name** to store the modified configuration, and review the connection URL.

8 Click **Finish**.

The modified configuration is stored, and can be run at any time. See "Running Import Configurations " on page 67.

# Deleting Import Configurations

➤ To delete import configurations:

1 Ensure that you have the connection information for the relational database on which the staging tables reside.

2 From **Task Areas**, select **Manage Model**, then **Import Staging Tables**.

The Import page is displayed.

3 Select the configuration to be deleted, and click the Delete button  .

A message asks you to confirm the deletion.

4 Click **Yes**.

The configuration is deleted.

# Running Import Configurations

After an import configuration has been created, you can run the configuration to import data into the application.

➤ To run an import configuration:

1 Ensure that you have the connection information for the relational database on which the staging tables reside.

2 From **Task Areas**, select **Manage Model**, and then **Import Staging Tables**

The Import page is displayed.

3 From the **Import Configurations List**, select the import configuration that you want to run.

4 Under **Log Levels**, select the level of detail you want to record.

5 Click the Run button  .

The Import Data dialog box is displayed. All data connection details are listed.

6 Click **Next**.

7 Under **Configuration**, verify that the target database and selected staging tables are correct.

8 Click **Next**.

The Configuration Summary is displayed. The Configuration Name and connection URL are identified.

9 Select one of the following operations:

- Click **Run Now** to run the import immediately.
- Click **Run Later** to schedule a more convenient date and time to run the import.

10 Click **Finish**.

- If you selected **Run Now**, the import runs and populates the target application with the selected data.
- If you selected **Run Later**, the job is saved. To run the saved import job, select Job Process, then Manage Taskflow. From this screen, you can run the job, or schedule it to run at a more convenient date and time. See the *Oracle Hyperion Profitability and Cost Management User's Guide* for detailed instructions.

11 When the import is complete, verify the imported data. See "Verifying Imported Data" on page 68.

# Verifying Imported Data

After importing the data into the target database, you must verify that the data has been imported correctly and completely.

➤ To verify imported data:

1 In Oracle Enterprise Performance Management Workspace, Fusion Edition, select **Navigate**, then **Applications**, then **Profitability**, and then the model name to access the Profitability and Cost Management model.

2 In the target application, open the model, and review the data that was expected in the import.

For example, if you imported Stages, select **Manage Model**, then **Stages** and ensure the information for all stages is present and correct.

3 Review the Exceptions column in each imported staging table to determine whether there are any errors or warnings. See Appendix A, "Staging Tables" for a list of error messages for each table.

4 Correct any errors in the source staging tables, and then re-run the import.

Repeat step 1 through step 4 as required until no errors are generated during the import. For a listing of the error messages for each staging table, see the Appendix A, "Staging Tables".

# 6 Migrating Data Using EPM System Lifecycle Management

**In This Chapter**



## About Lifecycle Management

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

Lifecycle Management features:

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

In addition to providing the Lifecycle Management interface in Oracle's Hyperion® 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.

**Note:** If you are using Lifecycle Management to import large models, the import may take longer to process than the time specified in the default timeout settings on Oracle WebLogic Server or the Oracle Application Server. To work around this issue, reset the default as described in "Resetting Default Timeout Settings " on page 70.

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

Using Lifecycle Management, Profitability and Cost Management models can be exported from the application server to file storage. The model of the application is exported to the file system as one artifact — model. The model artifact contains the following information:

- Driver definitions
- Stage definitions
- POV definitions
- Driver selections
- Assignments
- Assignment Rules
- Application preferences.

The following prerequisites must be met, whether exporting or importing the model:

- The application must be deployed from Oracle Hyperion EPM Architect, Fusion Edition to Profitability and Cost Management, and all dimensions must be available.
- The model must exist.

**Note:** You can import a previously exported model into a new, blank Profitability and Cost Management application, or into an existing application that contains model data. If it is an existing application, the existing model is deleted, and the new model is imported.

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

# Resetting Default Timeout Settings

If you are using Lifecycle Management to import large models, the import may take longer to process than the time specified in the default timeout settings on Oracle WebLogic Server or the Oracle Application Server. To work around this issue, you must reset the default timeout settings the appropriate server:

- "Default Timeout for WebLogic" on page 71
- "Default Timeout for Oracle Application Server (OAS)" on page 71

## Default Timeout for WebLogic

➤ To modify the default timeout for WebLogic Server:

1. Navigate to **HYPERION_HOME\common\httpServers\Apache\2.0.59\conf\HYSL-WebLogic.conf**.
2. In the configuration file, locate the section **LocationMatch/profitability**, and add the line **`WLIOTimeoutSecs 3000`**, as shown below:

   ```
   SetHandler weblogic-handlerPathTrimKeepAliveEnabled ONKeepAliveSecs
   20WLIOTimeoutSecs 3000WeblogicCluster servername: 6756
   ```

3. Navigate to **HYPERION_HOME\common\httpServers\Apache\2.0.59\conf\htppd.conf**.
4. Set **Timeout** to **3000**, as shown in the following text:

   ```
   # Timeout: The number of seconds before receives and sends time out.
   Timeout 3000
   ```

## Default Timeout for Oracle Application Server (OAS)

➤ To modify the default timeout for Oracle Application Server (OAS):

1. Navigate to **ORACLE_HOME\10.1.3.1\OracleAS_1\Apache\Apache\conf\httpd.conf**.
2. In the configuration file, locate the section **LocationMatch/profitability**, and add the line **`WLIOTimeoutSecs 3000`**, as shown below:

   ```
   SetHandler weblogic-handlerPathTrimKeepAliveEnabled ONKeepAliveSecs
   20WLIOTimeoutSecs 3000WeblogicCluster servername: 6756
   ```

3. Navigate to **HYPERION_HOME\common\httpServers\Apache\2.0.59\conf\htppd.conf**.
4. Set **Timeout** to **3000**, as shown in the following text:

   ```
   # Timeout: The number of seconds before receives and sends time out.
   Timeout 3000
   ```

# 7 Backing Up Profitability and Cost Management Components

There are several Profitability and Cost Management databases that need to be backed up on a regular basis to ensure the safety and integrity of your data:

- Operational Data Store
- Import Staging Tables
- Block Storage Option (BSO) Database

The frequency of backup is determined by the amount of change in the databases, and the general standards of your organization.

The following table, Table 9, "Profitability and Cost Management Database Backup Requirements," on page 73, lists the databases that must be backed up, their type, and suggested backup options.

**Note:** Lifecycle Management (LCM) cannot currently be used to back up the import staging area or the operational data store.

**Table 9** Profitability and Cost Management Database Backup Requirements

| Database | Database Type | Backup Options |
|---|---|---|
| Import Staging Area | Relational Database (RDB) | Standard backup techniques for any RDB can be used, including the following options:<br>• Database scripting<br>• Scheduler scripts<br>• Backup tools, such as TOAD<br>• Backup procedure from SQL Server or Oracle. |
| Operational Data Store | Relational Database (RDB) | Standard backup techniques for any RDB can be used, including the following options:<br>• Database scripting<br>• Scheduler scripts<br>• Backup tools, such as TOAD<br>• Backup procedure from SQL Server or Oracle. |

| Database | Database Type | Backup Options |
|---|---|---|
| Block Storage Option (BSO) Database and Assignment Rules | Essbase | Standard Essbase BSO backup techniques should be used. For detailed backup and recovery procedures,*Oracle Essbase Database Administrator's Guide*. |

For detailed backup and recovery procedures, see the *Oracle Hyperion Enterprise Performance Management System Backup and Recovery Guide*.

# A

# Staging Tables

**In This Appendix**



## Using Staging Tables

In order to import model data from relational databases into Profitability and Cost Management, you must create a set of staging tables in the source database to format the information for use in the application. Model data and metadata for the import are loaded into the staging tables in an Oracle or MS SQL Server database. To import data and metadata using these staging tables, see the *Oracle Hyperion Profitability and Cost Management User's Guide*.

The Profitability and Cost Management staging table schema has defined table names and contents.

When importing data, the sequence in which metadata or data is imported for the first time import of a complete model may be critical to the success of the transactions. For example, you must have dimensions and measures before you can define a point of view for a model. A listing of additional tables required for each staging table is displayed under the heading "Dependencies." The table dependencies apply only when importing the complete model.

A unique record ID is required for every staging table. This field is required, and is entered by the user, whether through an automatic program, script or manual entry.

The error messages for each table are listed in the Error Messages table below the associated schema table.

Use the schema tables in this appendix to create your staging tables:

- "Creating Database Views" on page 76
- "HPM_STG_STAGE" on page 76
- "HPM_STG_POV" on page 78

# Creating Database Views

An Administrator can create new database views in the system database that mirror the columns used in the staging tables, showing the model data that is stored in the system:

- Stages (HPM_STG_STAGE)
- POVs (HPM_STG_POV)
- Drivers (HPM_STG_DRIVER)
- Driver Selections (HPM_STG_DRIVER_SELECTION)
- Assignments (HPM_STG_ASSIGNMENT)
- Assignment Rules (HPM_STG_ASGN_RULE_SELECTION)

The database views are currently available using MS SQL Server and Oracle databases. You must run a separate query for each database view that you want to create.

➤ To create database views:

1. **Locate the Database Views in the following default location for your database type:**
   - `Hyperion Home/products/Profitability/database/Common/MSSQLServer`
   - `Hyperion Home/products/Profitability/database/Common/Oracle 2`
2. **Run a query to create a database view against your Profitability and Cost Management database.**
3. **Open the view to display all columns for the view, and associated data.**
4. **Repeat step 2 and step 3 for each view you want to create.**

# HPM_STG_STAGE

The HPM_STG_STAGE table provides a list of the stages within the model, and identifies the display order, prefix, associated dimensions and driver dimension for each stage.

**Dependencies:** None

**Table 10 HPM_STG_STAGE**

| Field | SQL Server DataType | Oracle Data Type | Required | Description |
|---|---|---|---|---|
| id | identity (1,1) | integer | Yes | Unique record ID |

| Field | SQL Server DataType | Oracle Data Type | Required | Description |
|---|---|---|---|---|
| name | nvarchar (80) | varchar2 (80) | Yes | Name of the model stage. |
| description | nvarchar (255) | varchar2 (255) | | Brief explanation of the purpose of the stage. |
| stage_order | integer | integer | | The sequence position in which the selected stage is to be used within the model while modeling (assignments and rules creation), in calc script generation and calculation. |
| stage_prefix | nvarchar (80) | varchar2 (80) | Yes | Differentiating prefix for the stage name. |
| dim1_name | nvarchar (80) | varchar2 (80) | Yes | Name of the first dimension included in the stage. |
| dim2_name | nvarchar (80) | varchar2 (80) | | Name of the second dimension included in the stage, if available. |
| dim3_name | nvarchar (80) | varchar2 (80) | | Name of the third dimension included in the stage, if available. |
| driver_dim_name | nvarchar (80) | varchar2 (80) | Yes | Name of the dimension selected as the driver for the stage. |
| Is_intra | nvarchar (1) | varchar2 (255) | Yes | Enter “Y” (Yes) to allow intrastage assignments within the model stage.<br>Default is “N” (No). This field is populated from the application. |
| last_upload_date | datetime | date | | The date and time of the last import. |
| exception | nvarchar (255) | varchar2 (255) | | Message detailing any errors that |

| Field | SQL Server DataType | Oracle Data Type | Required | Description |
|---|---|---|---|---|
| | | | | occurred during the import of this table.<br>See Table 11. |
| created_userid | nvarchar (32) | varchar2 (32) | Populated by the import program upon update. | The ID of the user who initiated the last import. |
| created_timestamp | datetime | date | Populated by the import program upon update. | The date and time of the the last import. This value is populated by the import program upon update. |
| modified_userid | nvarchar(32) | varchar2 (32) | Populated by the import program upon update. | The ID of the user who modified the last import. |
| modified_ timestamp | datetime | date | Populated by the import program upon update. | The date and time of the modifed import. |

**Table 11 HPM_STG_STAGE Error Messages**

| Error Message | Description |
|---|---|
| ERROR_NAME_NOT_ FOUND | No stage name exists. |
| ERROR_DIM_NAME_ NOT_FOUND | No stage dimension with the defined name exists in the dimension data. |

# HPM_STG_POV

The HPM_STG_POV table stores the states of each combination of dimension members included in a point of view (POV).

**Dependencies:** None

**Table 12 HPM_STG_POV**

| Field | SQL Server DataType | Oracle Data Type | Required | Description |
|---|---|---|---|---|
| id | integer identity (1,1) | integer | Yes | Unique record ID |
| pov_dim1_ member_name | nvarchar (80) | varchar2 (80) | Yes | Name of the first POV dimension member. |
| pov_dim2_ member_name | nvarchar (80) | varchar2 (80) | | Name of the second POV dimension member, if available. |

| Field | SQL Server DataType | Oracle Data Type | Required | Description |
|---|---|---|---|---|
| pov_dim3_ member_name | nvarchar (80) | varchar2 (80) | | Name of the third POV dimension member, if available.<br>Additional dimensions and member names may be added, as required. |
| pov_dim4_ member_name | nvarchar (80) | varchar2 (80) | | Name of the fourth POV dimension member, if available.<br>Additional dimensions and member names may be added, as required. |
| pov_state | nvarchar (80) | varchar2 (80) | Yes | The current status of the POV:<br>● Draft<br>● Published<br>● Archived |
| last_upload_date | datetime | date | | The date and time of the last import. |
| exception | nvarchar (255) | varchar2 (255) | | Message detailing any errors that occurred during the import of this table.<br>See Table 13. |
| created_userid | nvarchar (32) | varchar2 (32) | Populated by the import program upon update. | The ID of the user who initiated the last import. |
| created_timestamp | datetime | date | Populated by the import program upon update. | The date and time of the the last import. This value is populated by the import program upon update. |
| modified_userid | nvarchar(32) | varchar2 (32) | Populated by the import program upon update. | The ID of the user who modified the last import. |
| modified_ timestamp | datetime | date | Populated by the import program upon update. | The date and time of the modifed import. |

**Table 13** HPM_STG_POV Error Messages

| Error Message | Description |
|---|---|
| ERROR_POV_MEMBER_NOT_FOUND | No POV member with the defined name exists. |
| ERROR_POV_STATE_NOT_FOUND | No POV status with the defined name exists. |

# HPM_STG_DRIVER

The HPM_STG_DRIVER table provides details about the driver for the selected stage, including driver type, formula and layer.

**Dependencies:** None

**Table 14** HPM_STG_DRIVER

| Field | SQL Server DataType | Oracle Data Type | Required | Description |
|---|---|---|---|---|
| id | integer identity (1,1) | integer | Yes | Unique record ID |
| name | nvarchar (80) | varchar2 (80) | Yes | Name of the driver. |
| description | nvarchar (255) | varchar2 (255) | | Description of the purpose for the selected driver. |
| display_order | integer | integer | | Display position of the driver within the list of all drivers in the model. |
| driver_type | nvarchar (80) | varchar2 (80) | Yes | Type of driver:<br>● Even<br>● Simple<br>● Simple Weighted<br>● Simple Variable<br>● Weighted Variable<br>● Fixed and Variable<br>● Percentage<br>● Custom |
| fixed_member | nvarchar (80) | varchar2 (80) | | Member in the Measure dimension that corresponds to the standard driver measure "FixedDriverValue." |

| Field | SQL Server DataType | Oracle Data Type | Required | Description |
|---|---|---|---|---|
| fixed_location | nvarchar (30) | varchar2 (30) | | Location of the standard driver measure "FixedDriverValue." |
| rate_member | nvarchar (80) | varchar2 (80) | | Member in the Measure dimension that corresponds to the standard driver measure "Rate." |
| rate_location | nvarchar (30) | varchar2 (30) | | Location of the standard driver measure "Rate." |
| weight_member | nvarchar (80) | varchar2 (80) | | Member in the Measure dimension that corresponds to the standard driver measure "Weight." |
| weight_location | nvarchar (30) | varchar2 (30) | | Location of the standard driver measure "Weight." |
| volume_member | nvarchar (80) | varchar2 (80) | | Member in the Measure dimension that corresponds to the standard driver measure "Volume." |
| volume_location | nvarchar (30) | varchar2 (30) | | Location of the standard driver measure "Volume." |
| custom_formula | nvarchar (500) | varchar2 (500) | | Formula created for the driver by a user. This formula must be created using Essbase calculation script syntax. |
| cost_layer | nvarchar (30) | varchar2 (30) | | The driver layer is set to Cost layer, which contains the cost values for the model.<br>**Note:** Either cost_layer, revenue_layer, or both must be selected. |
| revenue_layer | nvarchar (30) | varchar2 (30) | | The driver layer is set to Revenue layer, which contains the revenue values for the model. |

| Field | SQL Server DataType | Oracle Data Type | Required | Description |
|---|---|---|---|---|
| | | | | **Note:** Either cost_ layer, revenue_layer, or both must be selected. |
| allow_idle_cost | nvarchar (1) | varchar2 (1) | | Enter appropriate value to determine whether idle costs are allowed for this driver:<br>● Enter "Y" (Yes) to allow idle cost for a driver.<br>● Enter "N" (No) to disable idle cost for a driver. This is the default. |
| last_upload_date | datetime | date | | The date and time of the last import. |
| exception | nvarchar (255) | varchar2 (255) | | Message detailing any errors that occurred during the import of this table.<br>See Table 15. |
| created_userid | nvarchar (32) | varchar2 (32) | Populated by the import program upon update. | The ID of the user who initiated the last import. |
| created_timestamp | datetime | date | Populated by the import program upon update. | The date and time of the last import. This value is populated by the import program upon update. |
| modified_userid | nvarchar(32) | varchar2 (32) | Populated by the import program upon update. | The ID of the user who modified the last import. |
| modified_ timestamp | datetime | date | Populated by the import program upon update. | The date and time of the modifed import. |

**Table 15 HPM_STG_DRIVER Error Messages**

| Error | Description |
|---|---|
| ERROR_NAME_NOT_FOUND | No driver name exists. |
| ERROR_DRIVER_LAYER_NOT_FOUND | No driver layer, or incorrect driver layer, has been selected for the driver. |

| Error | Description |
|---|---|
| ERROR_DRIVER_TYPE_NOT_FOUND | No driver type, or incorrect driver type, is associated with the selected driver. |

# HPM_STG_DRIVER_SELECTION

The HPM_STG_DRIVER_SELECTION table stores drivers selected for the intersections of the exact stage.

**Dependencies:**

- HPM_STG_DRIVER
- HPM_STG_STAGE
- HPM_STG_POV

**Table 16 HPM_STG_DRIVER_SELECTION**

| Field | SQL Server DataType | Oracle Data Type | Required | Description |
|---|---|---|---|---|
| id | integer identity (1,1) | integer | Yes | Unique record ID |
| pov_dim1_ member_name | nvarchar (80) | varchar2 (80) | Yes | Name of the first dimension member in the selected point of view. |
| pov_dim2_ member_name | nvarchar (80) | varchar2 (80) | | Name of the second dimension member in the point of view, if applicable. |
| pov_dim3_ member_name | nvarchar (80) | varchar2 (80) | | Name of the third dimension member in the point of view, if applicable. |
| pov_dim4_ member_name | nvarchar (80) | varchar2 (80) | | Name of the fourth dimension member in the point of view, if applicable. |
| layer_name | nvarchar (80) | varchar2 (80) | Yes | Name of the selected layer for the point of view:<br>● Cost (Default)<br>● Revenue |
| stage_name | nvarchar (80) | varchar2 (80) | Yes | Name of the model stage for the selected point of view. |

| Field | SQL Server DataType | Oracle Data Type | Required | Description |
|---|---|---|---|---|
| dim1_member_name | nvarchar (80) | varchar2 (80) | Yes, if driver dimension for the selected stage | Member name of the first dimension of the selected stage. |
| dim2_member_name | nvarchar (80) | varchar2 (80) | Yes, if driver dimension for the selected stage | Member name of the second dimension of the selected stage, if applicable. |
| dim3_member_name | nvarchar (80) | varchar2 (80) | Yes, if driver dimension for the selected stage | Member name of the third dimension of the selected stage, if applicable. |
| driver_name | nvarchar (80) | varchar2 (80) | Yes | Name of the driver for the selected intersection. |
| last_upload_date | datetime | date | | The date and time of the last import. |
| exception | nvarchar (255) | varchar2 (255) | | Message detailing any errors that occurred during the import of this table.<br>See Table 17. |
| created_userid | nvarchar (32) | varchar2 (32) | Populated by the import program upon update. | The ID of the user who initiated the last import. |
| created_timestamp | datetime | date | Populated by the import program upon update. | The date and time of the initiated the last import. This value is populated by the import program upon update. |
| modified_userid | nvarchar(32) | varchar2 (32) | Populated by the import program upon update. | The ID of the user who modified the last import. |
| modified_timestamp | datetime | date | Populated by the import program upon update. | The date and time of the modifed import. |

**Table 17 HPM_STG_DRIVER_SELECTION Error Messages**

| Error | Description |
|---|---|
| ERROR_POV_MEMBER_NOT_FOUND | One or more POV members are missing or do not match the Source POV Member data. |
| ERROR_STAGE_NOT_FOUND | No stage with the defined name exists. |

| Error | Description |
|---|---|
| ERROR_DRIVER_NOT_FOUND | No driver with the defined name exists. |
| ERROR_LAYER_NOT_FOUND | No layer with the defined name exists. |

# HPM_STG_ASSIGNMENT

The HPM_STG_ASSIGNMENT table provides details about each assignment, including the source stage, POV, driver layer, source dimension members, destination stage, and destination dimension members.

**Dependencies:**

- HPM_STG_POV
- HPM_STG_STAGE
- HPM_STG_DRIVER

**Table 18 HPM_STG_ASSIGNMENT**

| Field | SQL Server Data Type | Oracle Data Type | Required | Description |
|---|---|---|---|---|
| id | identity (1,1) | Identity | Yes | Unique record ID |
| pov_dim1_ member_name | nvarchar (80) | varchar2 (80) | Yes | Name of the first dimension member in the selected point of view. |
| pov_dim2_ member_name | nvarchar (80) | varchar2 (80) | | Name of the second dimension member in the point of view, if applicable. |
| pov_dim3_ member_name | nvarchar (80) | varchar2 (80) | | Name of the third dimension member in the point of view, if applicable. |
| pov_dim4_ member_name | nvarchar (80) | varchar2 (80) | | Name of the fourth dimension member in the point of view, if applicable. |
| layer_name | nvarchar (80) | varchar2 (80) | Yes | Name of the selected layer for the point of view:<br>● Cost (Default)<br>● Revenue<br>**Note:** Either the cost_layer, revenue_layer, |

| Field | SQL Server Data Type | Oracle Data Type | Required | Description |
|---|---|---|---|---|
| | | | | or both must be selected. |
| src_stage_name | nvarchar (80) | varchar2 (80) | Yes | Name of the source stage for the assignment. |
| src_dim1_member_name | nvarchar (80) | varchar2 (80) | Yes | Name of the first dimension member in the source stage. |
| src_dim2_member_name | nvarchar (80) | varchar2 (80) | Yes, if stage source/ destination stage have this dimension | Name of the second dimension member in the source stage, if available. |
| src_dim3_member_name | nvarchar (80) | varchar2 (80) | Yes, if stage source/ destination stage have this dimension | Name of the third dimension member in the source stage, if available. |
| dst_stage_name | nvarchar (80) | varchar2 (80) | Yes | Name of the destination stage for the assignment. |
| dst_dim1_member_name | nvarchar (80) | varchar2 (80) | Yes | Name of the first dimension member in the destination stage. |
| dst_dim2_member_name | nvarchar (80) | varchar2 (80) | | Name of the second dimension member in the destination stage, if available. |
| dst_dim3_member_name | nvarchar (80) | varchar2 (80) | | Name of the third dimension member in the destination stage, if available. |
| last_upload_date | datetime | date | | The date and time of the last import. |
| exception | nvarchar (255) | varchar2 (255) | | Message detailing any errors that occurred during the import of this table.<br>See Table 19. |
| created_userid | nvarchar (32) | varchar2 (32) | Populated by the import program upon update. | The ID of the user who initiated the last import. |
| created_timestamp | datetime | date | Populated by the import program upon update. | The date and time of the initiated the last import. This value is |

| Field | SQL Server Data Type | Oracle Data Type | Required | Description |
|---|---|---|---|---|
| | | | | populated by the import program upon update. |
| modified_userid | nvarchar(32) | varchar2 (32) | Populated by the import program upon update. | The ID of the user who modified the last import. |
| modified_ timestamp | datetime | date | Populated by the import program upon update. | The date and time of the modifed import. |

**Table 19 HPM_STG_ASSIGNMENT Error Messages**

| Error | Description |
|---|---|
| ERROR_POV_MEMBER_NOT_FOUND | One or more POV members are missing or there is a mismatch. |
| ERROR_STAGE_NOT_FOUND | No stagewith the defined name exists. |
| ERROR_LAYER_NOT_FOUND | No layer with the defined name exists. |

# HPM_STG_ASGN_RULE_SELECTION

The HPM_STG_ASGN_RULE_SELECTION table stores details about the source stage and dimension members for a rule controlling the assignments for the selected stage.

**Dependencies:**

- HPM_STG_POV
- HPM_STG_STAGE
- HPM_STG_DRIVER

**Table 20 HPM_STG_ASSGN_RULE_SELECTION**

| Field | SQL Server Data Type | Oracle Data Type | Required | Description |
|---|---|---|---|---|
| id | identity (1,1) | integer | Yes | Unique record ID |
| pov_dim1_ member_name | nvarchar (80) | varchar2 (80) | Yes | Name of the first dimension member in the selected point of view. |
| pov_dim2_ member_name | nvarchar (80) | varchar2 (80) | | Name of the second dimension member in the point of view, if applicable. |

| Field | SQL Server Data Type | Oracle Data Type | Required | Description |
|---|---|---|---|---|
| pov_dim3_ member_name | nvarchar (80) | varchar2 (80) | | Name of the third dimension member in the point of view, if applicable. |
| pov_dim4_ member_name | nvarchar (80) | varchar2 (80) | | Name of the fourth dimension member in the point of view, if applicable. |
| layer_name | nvarchar (80) | varchar2 (80) | Yes | Name of the selected layer for the point of view:<br>● Cost (Default)<br>● Revenue<br>**Note:** Either cost_layer, revenue_layer, or both is required. |
| src_stage_name | nvarchar (80) | varchar2 (80) | Yes | Name of the source stage for the assignment rule. |
| src_dim1_member_ name | nvarchar (80) | varchar2 (80) | Yes, if the source stage has the appropriate dimension. | Name of the first dimension member in the source stage. |
| src_dim2_member_ name | nvarchar (80) | varchar2 (80) | Yes, if the source stage has the appropriate dimension. | Name of the second dimension member in the source stage, if available. |
| src_dim3_member_ name | nvarchar (80) | varchar2 (80) | Yes, if the source stage has the appropriate dimension. | Name of the third dimension member in the source stage, if available. |
| rule_name | nvarchar (80) | varchar2 (80) | Yes | Name of the rule controlling the selected assignment. The rule must be present in the target database. |
| last_upload_date | date | date | | Date of the last import. |
| exception | nvarchar (255) | varchar2 (255) | | Message detailing any errors that occurred during the import of this table. |

| Field | SQL Server Data Type | Oracle Data Type | Required | Description |
|---|---|---|---|---|
| | | | | See Table 21. |
| created_timestamp | datetime | date | Populated by the import program upon update. | The date and time of the last import. |
| created_userid | nvarchar (32) | varchar2 (32) | Populated by the import program upon update. | ID of the user who initiated the last import. |
| modified_userid | nvarchar (32) | varchar2 (32) | Populated by the import program upon update. | ID of the user who initiated the reimport. |
| modified_ timestamp | date, time | date | Populated by the import program upon update. | The date and time of the reimport. |

**Table 21 HPM_STG_ASGN_RULE_SELECTION Error Messages**

| Error | Description |
|---|---|
| ERROR_POV_MEMBER_ NOT_FOUND | One or more POV members are missing or there is a mismatch. |
| ERROR_STAGE_NOT_ FOUND | No stage with the defined name exists. |
| ERROR_LAYER_NOT_ FOUND | No layer with the defined name exists. |
| ERROR_STAGE_RULE_ BACKWARD | The source stage display order is greater than, or equal to the rule stage display order. |
| ERROR_STAGE_RULE_ NOT_FOUND | No stage rule exists. |
| ERROR_DIMENSION_ MEMBER_COUNT | There is an incorrect dimension member count specified in the staging table.<br>For example, the Stage 2 has two dimensions, but there is only one dimension member provided in the staging table. |
| ERROR_NON_LEVEL_0_ DIMENSION MEMBER | A non-level-0 member has been specified in the staging table. All assignments must be set for level-0 members only. |

# B Essbase Naming Conventions

**In This Appendix**



## Generated Calculation Script Naming Conventions

When Oracle Hyperion Profitability and Cost Management, Fusion Edition generates Essbase scripts, the scripts are automatically named using specific conventions. The script name is created in the following format:

```
String scriptName = scriptSuffix + POV-identifier + Stage Order Number
+ "_" + index;
```

Each component of the script name.

- The script Suffix is based on the type of script. The following table displays the list of suffixes.

**Table 22 Calculation Script Suffixes**

| Script Type | Layer | |
|---|---|---|
| | Cost | Revenue |
| Inter-Stage Allocation | "a" | "r" |
| Inter-Stage Driver | "d" | "s" |
| Intra-Stage Allocation | "i" | "t" |
| Intra-Stage Driver | "j" | "u" |

- The POV-identifer is based on the POV ID, and may include up to 3 digits. A separate script is generated and identified for every POV.
- The Stage Order Number is the order number for the Source stage (for example, 1, 2, 3, and so on).
- If multiple scripts are generated due to script splitting, the _index display the numerical sequence of the scripts for the same type, POV , Stage and layer.

**Sample Script Names**

- `a3682_1.csc` represents an allocation calculation script for POV identifier 368, Source Stage 2, and the cost layer.
- `u4533_2.csc` represents an intra-stage driver calculation script for POV identifier 453, Source Stage 3, and the revenue layer.

# Essbase Naming Restrictions for Applications and Databases

When creating names for applications and databases, enter the name in the case in which you want the word displayed. The application or database name is created exactly as it is entered. If you enter the name as all capital letters (for instance, NEWAPP), Essbase does not automatically convert it to upper- and lowercase (for instance, Newapp).

**Note:** This list provides a partial set of the restrictions. For the complete list of all restrictions, Oracle recommends that you refer to the *Oracle Essbase Database Administrator's Guide:*

The following naming restrictions apply when you are naming applications and databases:

- Use no more than 8 bytes when naming non-Unicode-mode applications and databases; use no more than 30 characters when naming Unicode-mode applications and databases.
- Do not use spaces anywhere in the name.
- Do not use the following special characters in the name:
  - * asterisks
  - + plus signs
  - \ backslashes
  - ? question marks
  - [] brackets
  - " double quotation marks
  - : colons
  - ; semicolons
  - , commas
  - ‘ single quotation marks
  - = equal signs
  - / slashes
  - > greater than signs
  - tabs
  - < less than signs
  - | vertical bars

- . periods

- For aggregate storage databases only, do not use any of the following words as application or database names:
  - DEFAULT
  - LOG
  - METADATA
  - TEMP

## Essbase Naming Restrictions for Dimensions, Members and Aliases

When defining dimensional outlines, there are restricted characters that may not be used for naming dimensions, members and aliases. A list of the most common restricted characters is provided in this section; however, Oracle strongly suggests that you review the Essbase naming conventions described in the *Oracle Essbase Database Administrator's Guide* for a complete list.

**Note:** This list provides a partial set of the restrictions. For the complete list of all restrictions, Oracle recommends that you refer to the *Oracle Essbase Database Administrator's Guide:*

When naming dimensions, members and aliases, follow these naming restrictions:

- 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. To enable case-sensitivity, see "Setting Outline Properties" in the *Oracle Essbase Database Administrator's Guide*.
- Do not use HTML tags in member names, dimension names, aliases, and descriptions.
- Do not use quotation marks, brackets, backslashes, or tabs in a name. Brackets are permitted but not recommended in block storage outlines because they cause errors when converting to aggregate storage outlines.
- Do not use the following characters to begin dimension or member names:
  - at signs (@)
  - backslashes (\)
  - brackets ([])
  - commas (,)
  - dashes
  - hyphens
  - minus signs (-)
  - equal signs (=)

    - less than signs (<)
    - parentheses ( )
    - periods (.)
    - plus signs (+)
    - single quotation marks (')
    - quotation marks (")
    - underscores (_)
    - vertical bars(|)
- Do not place spaces at the beginning or end of names, as they are ignored by Oracle Essbase.
- Do not use forward slashes in member names.
- For time periods in custom calendars, do not use spaces in prefixes.
- Do not use any of the following items as dimension or member names.
    - Calculation script commands, operators or keywords. For a complete list of commands, see the *Oracle Essbase Database Administrator's Guide*.
    - Report writer commands
    - Function names and function arguments
    - If Dynamic Time Series is enabled, do not use History, Year, Season, Period, Quarter, Month, Week, or Day.
    - Names of other dimensions and members (unless the member is shared), generation names, level names, and aliases in the database.
- Do not use any of the following words:
    - ALL
    - AND
    - ASSIGN
    - AVERAGE
    - CALC
    - CALCMBR
    - COPYFORWARD
    - CROSSDIM
    - CURMBRNAME
    - DIM
    - DIMNAME
    - DIV
    - DYNAMIC
    - EMPTYPARM
    - EQ

- EQOP
- EXCEPT
- EXP
- EXPERROR
- FLOAT
- FUNCTION
- GE
- GEN
- GENRANGE
- GROUP
- GT
- ID
- IDERROR
- INTEGER
- LE
- LEVELRANGE
- LOOPBLOCK
- LOOPPARMS
- LT
- MBR
- MBRNAME
- MBRONLY
- MINUS
- MISSING
- MUL
- MULOP
- NE
- NON
- NONINPUT
- NOT
- OR
- PAREN
- PARENPARM
- PERCENT
- PLUS

- RELOP
- SET
- SKIPBOTH
- SKIPMISSING
- SKIPNONE
- SKIPZERO
- TO
- TOLOCALRATE
- TRAILMISSING
- TRAILSUM
- UMINUS
- UPPER
- VARORXMBR
- XMBRONLY
- $$UNIVERSE$$
- #MISSING
- #MI

# Glossary

**!** *See bang character (!).*

**#MISSING** *See missing data (#MISSING).*

**access permissions** A set of operations that a user can perform on a resource.

**accessor** Input and output data specifications for data mining algorithms.

**account blocking** The process by which accounts accept input data in the consolidated file. Blocked accounts do not receive their value through the additive consolidation process.

**account eliminations** Accounts which have their values set to zero in the consolidated file during consolidation.

**account type** How an account's value flows over time, and its sign behavior. Account type options can include expense, income, asset, liability, and equity.

**accountability map** A visual, hierarchical representation of the responsibility, reporting, and dependency structure of the accountability teams (also known as critical business areas) in an organization.

**accounts dimension** A dimension type that makes accounting intelligence available. Only one dimension can be defined as Accounts.

**active service** A service whose Run Type is set to Start rather than Hold.

**activity-level authorization** Defines user access to applications and the types of activities they can perform on applications, independent of the data that will be operated on.

**ad hoc report** An online analytical query created on-the-fly by an end user.

**adapter** Software that enables a program to integrate with data and metadata from target and source systems.

**adaptive states** Interactive Reporting Web Client level of permission.

**adjustment** *See journal entry (JE).*

**Advanced Relational Access** The integration of a relational database with an Essbase multidimensional database so that all data remains in the relational database and is mapped to summary-level data residing in the Essbase database.

**agent** An Essbase server process that starts and stops applications and databases, manages connections from users, and handles user-access security. The agent is referred to as ESSBASE.EXE.

**aggregate cell** A cell comprising several cells. For example, a data cell that uses Children(Year) expands to four cells containing Quarter 1, Quarter 2, Quarter 3, and Quarter 4 data.

**aggregate function** A type of function, such as sum or calculation of an average, that summarizes or performs analysis on data.

**aggregate limit** A limit placed on an aggregated request line item or aggregated metatopic item.

**aggregate storage database** The database storage model designed to support large-scale, sparsely distributed data which is categorized into many, potentially large dimensions. Upper level members and formulas are dynamically calculated, and selected data values are aggregated and stored, typically with improvements in overall aggregation time.

**aggregate view** A collection of aggregate cells based on the levels of the members within each dimension. To reduce calculation time, values are pre-aggregated and stored as aggregate views. Retrievals start from aggregate view totals and add up from there.

**aggregation** The process of rolling up and storing values in an aggregate storage database; the stored result of the aggregation process.

**aggregation script** In aggregate storage databases only, a file that defines a selection of aggregate views to be built into an aggregation.

**alias** An alternative name. For example, for a more easily identifiable column descriptor you can display the alias instead of the member name.

**alias table** A table that contains alternate names for members.

**alternate hierarchy** A hierarchy of shared members. An alternate hierarchy is based upon an existing hierarchy in a database outline, but has alternate levels in the dimension. An alternate hierarchy allows the same data to be seen from different points of view.

**ancestor** A branch member that has members below it. For example, the members Qtr2 and 2006 are ancestors of the member April.

**appender** A Log4j term for destination.

**application** (1) A software program designed to run a specific task or group of tasks such as a spreadsheet program or database management system. (2) A related set of dimensions and dimension members that are used to meet a specific set of analytical and/or reporting requirements.

**application currency** The default reporting currency for the application.

**area** A predefined set of members and values that makes up a partition.

**arithmetic data load** A data load that performs operations on values in the database, such as adding 10 to each value.

**artifact** An individual application or repository item; for example, scripts, forms, rules files, Interactive Reporting documents, and financial reports. Also known as an object.

**assemblies** Installation files for EPM System products or components.

**asset account** An account type that stores values that represent a company's assets.

**assignment** The association of a source and destination in the allocation model that controls the direction of allocated costs or revenue flow within Profitability and Cost Management.

**attribute** Characteristic of a dimension member. For example, Employee dimension members may have attributes of Name, Age, or Address. Product dimension members can have several attributes, such as a size and flavor.

**attribute association** A relationship in a database outline whereby a member in an attribute dimension describes a characteristic of a member of its base dimension. For example, if product 100-10 has a grape flavor, the product 100-10 has the Flavor attribute association of grape. Thus, the 100-10 member of the Product dimension is associated with the Grape member of the Flavor attribute dimension.

**Attribute Calculations dimension** A system-defined dimension that performs these calculation operations on groups of members: Sum, Count, Avg, Min, and Max. This dimension is calculated dynamically and is not visible in the database outline. For example, using the Avg member, you can calculate the average sales value for Red products in New York in January.

**attribute dimension** A type of dimension that enables analysis based on the attributes or qualities of dimension members.

**attribute reporting** A reporting process based on the attributes of the base dimension members. *See also base dimension*.

**attribute type** A text, numeric, Boolean, date, or linked-attribute type that enables different functions for grouping, selecting, or calculating data. For example, because the Ounces attribute dimension has the type numeric, the number of ounces specified as the attribute of each product can be used to calculate the profit per ounce for that product.

**authentication** Verification of identity as a security measure. Authentication is typically based on a user name and password. Passwords and digital signatures are forms of authentication.

**authentication service** A core service that manages one authentication system.

**auto-reversing journal** A journal for entering adjustments that you want to reverse in the next period.

**automated stage** A stage that does not require human intervention, for example, a data load.

**axis** (1) A straight line that passes through a graphic used for measurement and categorization. (2) A report aspect used to arrange and relate multidimensional data, such as filters, pages, rows, and columns. For example, for a data query in Simple Basic, an axis can define columns for values for Qtr1, Qtr2, Qtr3, and Qtr4. Row data would be retrieved with totals in the following hierarchy: Market, Product.

**backup** A duplicate copy of an application instance.

**balance account** An account type that stores unsigned values that relate to a particular point in time.

**balanced journal** A journal in which the total debits equal the total credits.

**bang character (!)** A character that terminates a series of report commands and requests information from the database. A report script must be terminated with a bang character; several bang characters can be used within a report script.

**bar chart** A chart that can consist of one to 50 data sets, with any number of values assigned to each data set. Data sets are displayed as groups of corresponding bars, stacked bars, or individual bars in separate rows.

**base currency** The currency in which daily business transactions are performed.

**base dimension** A standard dimension that is associated with one or more attribute dimensions. For example, assuming products have flavors, the Product dimension is the base dimension for the Flavors attribute dimension.

**base entity** An entity at the bottom of the organization structure that does not own other entities.

**batch calculation** Any calculation on a database that is done in batch; for example, a calculation script or a full database calculation. Dynamic calculations are not considered to be batch calculations.

**batch file** An operating system file that can call multiple ESSCMD scripts and run multiple sessions of ESSCMD. On Windows-based systems, batch files have BAT file extensions. On UNIX, batch files are written as a shell script.

**batch loader** An FDM component that enables the processing of multiple files.

**batch POV** A collection of all dimensions on the user POV of every report and book in the batch. While scheduling the batch, you can set the members selected on the batch POV.

**batch processing mode** A method of using ESSCMD to write a batch or script file that can be used to automate routine server maintenance and diagnostic tasks. ESSCMD script files can execute multiple commands and can be run from the operating system command line or from within operating system batch files. Batch files can be used to call multiple ESSCMD scripts or run multiple instances of ESSCMD.

**block** The primary storage unit which is a multidimensional array representing the cells of all dense dimensions.

**block storage database** The Essbase database storage model categorizing and storing data based on the sparsity of data values defined in sparse dimensions. Data values are stored in blocks, which exist only for sparse dimension members for which there are values.

**Blocked Account** An account that you do not want calculated in the consolidated file because you want to enter it manually.

**book** A container that holds a group of similar Financial Reporting documents. Books may specify dimension sections or dimension changes.

**book POV** The dimension members for which a book is run.

**bookmark** A link to a reporting document or a Web site, displayed on a personal page of a user. The two types of bookmarks are My Bookmarks and image bookmarks.

**bounding rectangle** The required perimeter that encapsulates the Interactive Reporting document content when embedding Interactive Reporting document sections in a personal page, specified in pixels for height and width or row per page.

**broadcast message** A simple text message sent by an administrator to a user who is logged on to a Planning application. The message displays information to the user such as system availability, notification of application refresh, or application backups.

**budget administrator** A person responsible for setting up, configuring, maintaining, and controlling an application. Has all application privileges and data access permissions.

**build method** A method used to modify database outlines. Choice of a build method is based on the format of data in data source files.

**business process** A set of activities that collectively accomplish a business objective.

**business rules** Logical expressions or formulas that are created within an application to produce a desired set of resulting values.

**cache** A buffer in memory that holds data temporarily.

**calc script** A set of commands that define how a database is consolidated or aggregated. A calculation script may also contain commands that specify allocation and other calculation rules separate from the consolidation process.

**calculated member in MaxL DML** A member designed for analytical purposes and defined in the optional WITH section of a MaxL DML query.

**calculated member in MaxL DML** A member designed for analytical purposes and defined in the optional WITH section of a MaxL DML query.

**calculation** The process of aggregating data, or of running a calculation script on a database.

**Calculation Manager** A calculation module with which Planning, Financial Management, and Essbase users can design, validate, and administer business rules in a graphical environment.

**calculation status** A consolidation status that indicates that some values or formula calculations have changed. You must reconsolidate to get the correct values for the affected entity.

**calendar** User-defined time periods and their relationship to each other. Q1, Q2, Q3, and Q4 comprise a calendar or fiscal year.

**cascade** The process of creating multiple reports for a subset of member values.

**Catalog pane** Displays a list of elements available to the active section. If Query is the active section, a list of database tables is displayed. If Pivot is the active section, a list of results columns is displayed. If Dashboard is the active section, a list of embeddable sections, graphic tools, and control tools are displayed.

**categories** Groupings by which data is organized. For example, Month.

**cause and effect map** Depicts how the elements that form your corporate strategy relate and how they work together to meet your organization's strategic goals. A Cause and Effect map tab is automatically created for each Strategy map.

**CDF** *See custom-defined function (CDF).*

**CDM** *See custom-defined macro (CDM).*

**cell** (1) The data value at the intersection of dimensions in a multidimensional database; the intersection of a row and a column in a worksheet. (2) A logical group of nodes belonging to one administrative domain.

**cell note** A text annotation for a cell in an Essbase database. Cell notes are a type of LRO.

**CHANGED status** Consolidation status that indicates data for an entity has changed.

**chart** A graphical representation of spreadsheet data. The visual nature expedites analysis, color-coding, and visual cues that aid comparisons.

**chart template** A template that defines the metrics to display in Workspace charts.

**child** A member with a parent above it in the database outline.

**choice list** A list of members that a report designer can specify for each dimension when defining the report's point of view. A user who wants to change the point of view for a dimension that uses a choice list can select only the members specified in that defined member list or those members that meet the criteria defined in the function for the dynamic list.

**clean block** A data block that where the database is fully calculated, if a calculation script calculates all dimensions at once, or if the SET CLEARUPDATESTATUS command is used in a calculation script.

**cluster** An array of servers or databases that behave as a single resource which share task loads and provide failover support; eliminates one server or database as a single point of failure in a system.

**clustered bar charts** Charts in which categories are viewed side-by-side; useful for side-by-side category analysis; used only with vertical bar charts.

**code page** A mapping of bit combinations to a set of text characters. Different code pages support different sets of characters. Each computer contains a code page setting for the character set requirements of the language of the computer user. In the context of this document, code pages map characters to bit combinations for non-Unicode encodings. *See also encoding*.

**column** A vertical display of information in a grid or table. A column can contain data from one field, derived data from a calculation, or textual information.

**committed access** An Essbase Kernel Isolation Level setting that affects how Essbase handles transactions. Under committed access, concurrent transactions hold long-term write locks and yield predictable results.

**computed item** A virtual column (as opposed to a column that is physically stored in the database or cube) that can be calculated by the database during a query, or by Interactive Reporting Studio in the Results section. Computed items are calculations of data based on functions, data items, and operators provided in the dialog box and can be included in reports or reused to calculate other data.

**configuration file** The security platform relies on XML documents to be configured by the product administrator or software installer. The XML document must be modified to indicate meaningful values for properties, specifying locations and attributes pertaining to the corporate authentication scenario.

**connection file** *See Interactive Reporting connection file (.oce)*.

**consolidated file (Parent)** A file into which all of the business unit files are consolidated; contains the definition of the consolidation.

**consolidation** The process of aggregating data from dependent entities to parent entities. For example, if the dimension Year consists of the members Qtr1, Qtr2, Qtr3, and Qtr4, its consolidation is Year.

**consolidation file (*.cns)** The consolidation file is a graphical interface that enables you to add, delete or move Strategic Finance files in the consolidation process using either a Chart or Tree view. It also enables you to define and modify the consolidation.

**consolidation rule** Identifies the rule that is executed during the consolidation of the node of the hierarchy. This rule can contain customer specific formulas appropriate for the correct consolidation of parent balances. Elimination processing can be controlled within these rules.

**content** Information stored in the repository for any type of file.

**content browser** A Component that allows users to Browse and select content to be placed in a Workspace Page .

**context variable** A variable that is defined for a particular task flow to identify the context of the taskflow instance.

**contribution** The value added to a parent from a child entity. Each child has a contribution to its parent.

**controls group** Used in FDM to maintain and organize certification and assessment information, especially helpful for meeting Sarbanes-Oxley requirements.

**conversion rate** *See exchange rate*.

**cookie** A segment of data placed on your computer by a Web site.

**correlated subqueries** Subqueries that are evaluated once for every row in the parent query; created by joining a topic item in the subquery with a topic in the parent query.

**critical business area (CBA)** An individual or a group organized into a division, region, plant, cost center, profit center, project team, or process; also called accountability team or business area.

**critical success factor (CSF)** A capability that must be established and sustained to achieve a strategic objective; owned by a strategic objective or a critical process and is a parent to one or more actions.

**crosstab reporting** Categorizes and summarizes data in table format. The table cells contain summaries of the data that fit within the intersecting categories. For example, a crosstab report of product sales information could show size attributes, such as Small and Large, as column headings and color attributes, such as Blue and Yellow, as row headings. The cell in the table where Large and Blue intersect could contain the total sales of all Blue products that are sized Large.

**cube** A block of data that contains three or more dimensions. An Essbase database is a cube.

**cube deployment** In Essbase Studio, the process of setting load options for a model to build an outline and load data into an Essbase application and database.

**cube schema** In Essbase Studio, the metadata elements, such as measures and hierarchies, representing the logical model of a cube.

**currency conversion** A process that converts currency values in a database from one currency into another. For example, to convert one U. S. dollar into the European euro, the exchange rate (for example, 0.923702) is multiplied with the dollar (1* 0.923702). After conversion, the European euro amount is .92.

**Currency Overrides** In any input period, the selected input method can be overridden to enable input of that period's value as Default Currency/Items. To override the input method, enter a pound sign (#) either before or after the number.

**currency partition** A dimension type that separates local currency members from a base currency, as defined in an application. Identifies currency types, such as Actual, Budget, and Forecast.

**custom calendar** Any calendar created by an administrator.

**custom dimension** A dimension created and defined by users. Channel, product, department, project, or region could be custom dimensions.

**custom property** A property of a dimension or dimension member that is created by a user.

**custom report** A complex report from the Design Report module, composed of any combination of components.

**custom-defined function (CDF)** Essbase calculation functions developed in Java and added to the standard Essbase calculation scripting language using MaxL. *See also custom-defined macro (CDM).*

**custom-defined macro (CDM)** Essbase macros written with Essbase calculator functions and special macro functions. Custom-defined macros use an internal Essbase macro language that enables the combination of calculation functions and they operate on multiple input parameters. *See also custom-defined function (CDF).*

**cycle through** To perform multiple passes through a database while calculating it.

**dashboard** A collection of metrics and indicators that provide an interactive summary of your business. Dashboards enable you to build and deploy analytic applications.

**data cache** A buffer in memory that holds uncompressed data blocks.

**data cell** *See cell.*

**data file cache** A buffer in memory that holds compressed data (PAG) files.

**data form** A grid display that enables users to enter data into the database from an interface such as a Web browser, and to view and analyze data or related text. Certain dimension member values are fixed, giving users a specific view into the data.

**data function** That computes aggregate values, including averages, maximums, counts, and other statistics, that summarize groupings of data.

**data load location** In FDM, a reporting unit responsible for submitting source data into the target system. Typically, there is one FDM data load location for each source file loaded to the target system.

**data load rules** A set of criteria that determines how to load data from a text-based file, a spreadsheet, or a relational data set into a database.

**data lock** Prevents changes to data according to specified criteria, such as period or scenario.

**data mining** The process of searching through an Essbase database for hidden relationships and patterns in a large amount of data.

**data model** A representation of a subset of database tables.

**data value** *See cell.*

**database connection** File that stores definitions and properties used to connect to data sources and enables database references to be portable and widely used.

**date measure** In Essbase, a member tagged as "Date" in the dimension where measures are represented. The cell values are displayed as formatted dates. Dates as measures can be useful for types of analysis that are difficult to represent using the Time dimension. For example, an application may need to track acquisition dates for a series of capital assets, but the acquisition dates span too large a period to allow for feasible Time dimension modeling. *See also typed measure.*

**Default Currency Units** Define the unit scale of data. For example, if you select to define your analysis in Thousands, and enter "10", this is interpreted as "10,000".

**dense dimension** In block storage databases, a dimension likely to contain data for every combination of dimension members. For example, time dimensions are often dense because they can contain all combinations of all members. *Contrast with sparse dimension.*

**dependent entity** An entity that is owned by another entity in the organization.

**derived text measure** In Essbase Studio, a text measure whose values are governed by a predefined rule expressed as a range. For example, a derived text measure, called "Sales Performance Index," based on a measure Sales, could consist of the values "High," "Medium," and "Low." This derived text measure is defined to display "High," "Medium," and "Low" depending on the range in which the corresponding sales values fall. *See also text measure.*

**descendant** Any member below a parent in the database outline. In a dimension that includes years, quarters, and months, the members Qtr2 and April are descendants of the member Year.

**Design Report** An interface in Web Analysis Studio for designing custom reports, from a library of components.

**destination** (1) For Business Rules and Calculation Manager, an intersection within the database where allocated values are stored. (2) Within a Profitability and Cost Management assignment, the receiving point for allocated values.

**destination currency** The currency to which balances are converted. You enter exchange rates and convert from the source currency to the destination currency. For example, when you convert from EUR to USD, the destination currency is USD.

**detail chart** A chart that provides the detailed information that you see in a Summary chart. Detail charts appear in the Investigate Section in columns below the Summary charts. If the Summary chart shows a Pie chart, then the Detail charts below represent each piece of the pie.

**dimension** A data category used to organize business data for retrieval and preservation of values. Dimensions usually contain hierarchies of related members grouped within them. For example, a Year dimension often includes members for each time period, such as quarters and months.

**dimension build** The process of adding dimensions and members to an Essbase outline.

**dimension build rules** Specifications, similar to data load rules, that Essbase uses to modify an outline. The modification is based on data in an external data source file.

**dimension tab** In the Pivot section, the tab that enables you to pivot data between rows and columns.

**dimension table** (1) A table that includes numerous attributes about a specific business process. (2) In Essbase Integration Services, a container in the OLAP model for one or more relational tables that define a potential dimension in Essbase.

**dimension type** A dimension property that enables the use of predefined functionality. Dimensions tagged as time have a predefined calendar functionality.

**dimensionality** In MaxL DML, the represented dimensions (and the order in which they are represented) in a set. For example, the following set consists of two tuples of the same dimensionality because they both reflect the dimensions (Region, Year): { (West, Feb), (East, Mar) }

**direct rate** A currency rate that you enter in the exchange rate table. The direct rate is used for currency conversion. For example, to convert balances from JPY to USD, In the exchange rate table, enter a rate for the period/scenario where the source currency is JPY and the destination currency is USD.

**dirty block** A data block containing cells that have been changed since the last calculation. Upper level blocks are marked as dirty if their child blocks are dirty (that is, they have been updated).

**display type** One of three Web Analysis formats saved to the repository: spreadsheet, chart, and pinboard.

**dog-ear** The flipped page corner in the upper right corner of the chart header area.

**domain** In data mining, a variable representing a range of navigation within data.

**drill-down** Navigation through the query result set using the dimensional hierarchy. Drilling down moves the user perspective from aggregated data to detail. For example, drilling down can reveal hierarchical relationships between years and quarters or quarters and months.

**drill-through** The navigation from a value in one data source to corresponding data in another source.

**driver** A driver is an allocation method that describes the mathematical relationship between the sources that utilize the driver, and the destinations to which those sources allocate cost or revenue. For Business Modeling, see also Cost Driver and Activity Driver.

**duplicate alias name** A name that occurs more than once in an alias table and that can be associated with more than one member in a database outline. Duplicate alias names can be used with duplicate member outlines only.

**duplicate member name** The multiple occurrence of a member name in a database, with each occurrence representing a different member. For example, a database has two members named "New York." One member represents New York state and the other member represents New York city.

**duplicate member outline** A database outline containing duplicate member names.

**Dynamic Calc and Store members** A member in a block storage outline that Essbase calculates only upon the first retrieval of the value. Essbase then stores the calculated value in the database. Subsequent retrievals do not require calculating.

**Dynamic Calc members** A member in a block storage outline that Essbase calculates only at retrieval time. Essbase discards calculated values after completing the retrieval request.

**dynamic calculation** In Essbase, a calculation that occurs only when you retrieve data on a member that is tagged as Dynamic Calc or Dynamic Calc and Store. The member's values are calculated at retrieval time instead of being precalculated during batch calculation.

**dynamic hierarchy** In aggregate storage database outlines only, a hierarchy in which members are calculated at retrieval time.

**dynamic member list** A system-created named member set that is based on user-defined criteria. The list is refreshed automatically whenever it is referenced in the application. As dimension members are added and deleted, the list automatically reapplies the criteria to reflect the changes.

**dynamic reference** A pointer in the rules file to header records in a data source.

**dynamic report** A report containing data that is updated when you run the report.

**Dynamic Time Series** A process that performs period-to-date reporting in block storage databases.

**dynamic view account** An account type indicating that account values are calculated dynamically from the data that is displayed.

**Eliminated Account** An account that does not appear in the consolidated file.

**elimination** The process of zeroing out (eliminating) transactions between entities within an organization.

**employee** A user responsible for, or associated with, specific business objects. Employees need not work for an organization; for example, they can be consultants. Employees must be associated with user accounts for authorization purposes.

**encoding** A method for mapping bit combinations to characters for creating, storing, and displaying text. Each encoding has a name; for example, UTF-8. Within an encoding, each character maps to a specific bit combination; for example, in UTF-8, uppercase A maps to HEX41. *See also code page* and *locale*.

**ending period** A period enabling you to adjust the date range in a chart. For example, an ending period of "month", produces a chart showing information through the end of the current month.

**Enterprise View** An Administration Services feature that enables management of the Essbase environment from a graphical tree view. From Enterprise View, you can operate directly on Essbase artifacts.

**entity** A dimension representing organizational units. Examples: divisions, subsidiaries, plants, regions, products, or other financial reporting units.

**Equity Beta** The riskiness of a stock, measured by the variance between its return and the market return, indicated by an index called "beta". For example, if a stock's return normally moves up or down 1.2% when the market moves up or down 1%, the stock has a beta of 1.2.

**essbase.cfg** An optional configuration file for Essbase. Administrators may edit this file to customize Essbase Server functionality. Some configuration settings may also be used with Essbase clients to override Essbase Server settings.

**EssCell** A function entered into a cell in Essbase Spreadsheet Add-in to retrieve a value representing an intersection of specific Essbase database members.

**ESSCMD** A command-line interface for performing Essbase operations interactively or through batch script files.

**ESSLANG** The Essbase environment variable that defines the encoding used to interpret text characters. *See also encoding*.

**ESSMSH** *See MaxL Shell*.

**exceptions** Values that satisfy predefined conditions. You can define formatting indicators or notify subscribing users when exceptions are generated.

**exchange rate** A numeric value for converting one currency to another. For example, to convert 1 USD into EUR, the exchange rate of 0.8936 is multiplied with the U.S. dollar. The European euro equivalent of $1 is 0.8936.

**exchange rate type** An identifier for an exchange rate. Different rate types are used because there may be multiple rates for a period and year. Users traditionally define rates at period end for the average rate of the period and for the end of the period. Additional rate types are historical rates, budget rates, forecast rates, and so on. A rate type applies to one point in time.

**expense account** An account that stores periodic and year-to-date values that decrease net worth if they are positive.

**Extensible Markup Language (XML)** A language comprising a set of tags used to assign attributes to data that can be interpreted between applications according to a schema.

**external authentication** Logging on to Oracle's Hyperion applications with user information stored outside the applications, typically in a corporate directory such as MSAD or NTLM.

**externally triggered events** Non-time-based events for scheduling job runs.

**Extract, Transform, and Load (ETL)** Data source-specific programs for extracting data and migrating it to applications.

**extraction command** An Essbase reporting command that handles the selection, orientation, grouping, and ordering of raw data extracted from a database; begins with the less than (<) character.

**fact table** The central table in a star join schema, characterized by a foreign key and elements drawn from a dimension table. This table typically contains numeric data that can be related to all other tables in the schema.

**Favorites gadget** Contains links to Reporting and Analysis documents and URLs.

**field** An item in a data source file to be loaded into an Essbase database.

**file delimiter** Characters, such as commas or tabs, that separate fields in a data source.

**filter** A constraint on data sets that restricts values to specific criteria; for example, to exclude certain tables, metadata, or values, or to control access.

**flow account** An unsigned account that stores periodic and year-to-date values.

**folder** A file containing other files for the purpose of structuring a hierarchy.

**footer** Text or images at the bottom of report pages, containing dynamic functions or static text such as page numbers, dates, logos, titles or file names, and author names.

**format** Visual characteristics of documents or report objects.

**format string** In Essbase, a method for transforming the way cell values are displayed.

**formula** A combination of operators, functions, dimension and member names, and numeric constants calculating database members.

**frame** An area on the desktop. There are two main areas: the navigation and Workspace frames.

**free-form grid** An object for presenting, entering, and integrating data from different sources for dynamic calculations.

**free-form reporting** Creating reports by entering dimension members or report script commands in worksheets.

**function** A routine that returns values or database members.

**gadget** Simple, specialized, lightweight applications that provide easy viewing of EPM content and enable access to core Reporting and Analysis functionality.

**genealogy data** Additional data that is optionally generated after allocation calculations. This data enables reporting on all cost or revenue flows from start to finish through all allocation steps.

**generation** A layer in a hierarchical tree structure that defines member relationships in a database. Generations are ordered incrementally from the top member of the dimension (generation 1) down to the child members. Use the unique generation name to identify a layer in the hierarchical tree structure.

**generic jobs** Non-SQR Production Reporting or non-Interactive Reporting jobs.

**global report command** A command in a running report script that is effective until replaced by another global command or the file ends.

**grid POV** A means for specifying dimension members on a grid without placing dimensions in rows, columns, or page intersections. A report designer can set POV values at the grid level, preventing user POVs from affecting the grid. If a dimension has one grid value, you put the dimension into the grid POV instead of the row, column, or page.

**group** A container for assigning similar access permissions to multiple users.

**GUI** Graphical user interface

**head up display** A mode that shows your loaded Smart Space desktop including the background image above your Windows desktop.

**highlighting** Depending on your configuration, chart cells or ZoomChart details may be highlighted, indicating value status: red (bad), yellow (warning), or green (good).

**Historical Average** An average for an account over a number of historical periods.

**holding company** An entity that is part of a legal entity group, with direct or indirect investments in all entities in the group.

**host** A server on which applications and services are installed.

**host properties** Properties pertaining to a host, or if the host has multiple Install_Homes, to an Install_Home. The host properties are configured from the CMC.

**Hybrid Analysis** An analysis mapping low-level data stored in a relational database to summary-level data stored in Essbase, combining the mass scalability of relational systems with multidimensional data.

**hyperlink** A link to a file, Web page, or an intranet HTML page.

**Hypertext Markup Language (HTML)** A programming language specifying how Web browsers display data.

**identity** A unique identification for a user or group in external authentication.

**image bookmarks** Graphic links to Web pages or repository items.

**IMPACTED status** Indicates changes in child entities consolidating into parent entities.

**implied share** A member with one or more children, but only one is consolidated, so the parent and child share a value.

**import format** In FDM, defines the structure of the source file which enables the loading of a source data file to an FDM data load location.

**inactive group** A group for which an administrator has deactivated system access.

**inactive service** A service suspended from operating.

**INACTIVE status** Indicates entities deactivated from consolidation for the current period.

**inactive user** A user whose account has been deactivated by an administrator.

**income account** An account storing periodic and year-to-date values that, if positive, increase net worth.

**index** (1) A method where Essbase uses sparse-data combinations to retrieve data in block storage databases. (2) The index file.

**index cache** A buffer containing index pages.

**index entry** A pointer to an intersection of sparse dimensions. Index entries point to data blocks on disk and use offsets to locate cells.

**index file** An Essbase file storing block storage data retrieval information, residing on disk, and containing index pages.

**index page** A subdivision in an index file. Contains pointers to data blocks.

**input data** Data loaded from a source rather than calculated.

**Install_Home** A variable for the directory where EPM System products are installed. Refers to one instance of an EPM System product when multiple applications are installed on the same computer.

**integration** Process that is run to move data between EPM System products using Shared Services. Data integration definitions specify the data moving between a source application and a destination application, and enable the data movements to be grouped, ordered, and scheduled.

**intelligent calculation** A calculation method tracking updated data blocks since the last calculation.

**Interactive Reporting connection file (.oce)** Files encapsulating database connection information, including: the database API (ODBC, SQL*Net, etc.), database software, the database server network address, and database user name. Administrators create and publish Interactive Reporting connection files (.oce).

**intercompany elimination** *See elimination*.

**intercompany matching** The process of comparing balances for pairs of intercompany accounts within an application. Intercompany receivables are compared to intercompany payables for matches. Matching accounts are used to eliminate intercompany transactions from an organization's consolidated totals.

**intercompany matching report** A report that compares intercompany account balances and indicates if the accounts are in, or out, of balance.

**interdimensional irrelevance** A situation in which a dimension does not intersect with other dimensions. Because the data in the dimension cannot be accessed from the non-intersecting dimensions, the non-intersecting dimensions are not relevant to that dimension.

**intersection** A unit of data representing the intersection of dimensions in a multidimensional database; also, a worksheet cell.

**intrastage assignment** Assignments in the financial flow that are assigned to objects within the same stage.

**introspection** A deep inspection of a data source to discover hierarchies based on the inherent relationships in the database. *Contrast with scraping*.

**Investigation** *See drill-through*.

**isolation level** An Essbase Kernel setting that determines the lock and commit behavior of database operations. Choices are: committed access and uncommitted access.

**iteration** A "pass" of the budget or planning cycle in which the same version of data is revised and promoted.

**Java Database Connectivity (JDBC)** A client-server communication protocol used by Java based clients and relational databases. The JDBC interface provides a call-level API for SQL-based database access.

**job output** Files or reports produced from running a job.

**jobs** Documents with special properties that can be launched to generate output. A job can contain Interactive Reporting, SQR Production Reporting, or generic documents.

**join** A link between two relational database tables or topics based on common content in a column or row. A join typically occurs between identical or similar items within different tables or topics. For example, a record in the Customer table is joined to a record in the Orders table because the Customer ID value is the same in each table.

**journal entry (JE)** A set of debit/credit adjustments to account balances for a scenario and period.

**JSP** Java Server Pages.

**KeyContacts gadget** Contains a group of Smart Space users and provides access to Smart Space Collaborator. For example, you can have a KeyContacts gadget for your marketing team and another for your development team.

**latest** A Spreadsheet key word used to extract data values from the member defined as the latest time period.

**layer** (1) The horizontal location of members in a hierarchical structure, specified by generation (top down) or level (bottom up). (2) Position of objects relative to other objects. For example, in the Sample Basic database, Qtr1 and Qtr4 are in the same layer, so they are also in the same generation, but in a database with a ragged hierarchy, Qtr1 and Qtr4 might not be in same layer, though they are in the same generation.

**layout area** Used to designate an area on a Workspace Page where content can be placed.

**legend box** A box containing labels that identify the data categories of a dimension.

**level** A layer in a hierarchical tree structure that defines database member relationships. Levels are ordered from the bottom dimension member (level 0) up to the parent members.

**level 0 block** A data block for combinations of sparse, level 0 members.

**level 0 member** A member that has no children.

**liability account** An account type that stores "point in time" balances of a company's liabilities. Examples of liability accounts include accrued expenses, accounts payable, and long term debt.

**life cycle management** The process of managing application information from inception to retirement.

**Lifecycle Management Utility** A command-line utility for migrating applications and artifacts.

**line chart** A chart that displays one to 50 data sets, each represented by a line. A line chart can display each line stacked on the preceding ones, as represented by an absolute value or a percent.

**line item detail** The lowest level of detail in an account.

**lineage** The relationship between different metadata elements showing how one metadata element is derived from one or more other metadata elements, ultimately tracing the metadata element to its physical source. In Essbase Studio, a lineage viewer displays the relationships graphically. *See also traceability*.

**link** (1) A reference to a repository object. Links can reference folders, files, shortcuts, and other links. (2) In a task flow, the point where the activity in one stage ends and another begins.

**link condition** A logical expression evaluated by the taskflow engine to determine the sequence of launching taskflow stages.

**linked data model** Documents that are linked to a master copy in a repository.

**linked partition** A shared partition that enables you to use a data cell to link two databases. When a user clicks a linked cell in a worksheet, Essbase opens a new sheet displaying the dimensions in the linked database. The user can then drill down those dimensions.

**linked reporting object (LRO)** A cell-based link to an external file such as cell notes, URLs, or files with text, audio, video, or pictures. (Only cell notes are supported for Essbase LROs in Financial Reporting.) *Contrast with local report object*.

**local currency** An input currency type. When an input currency type is not specified, the local currency matches the entity's base currency.

**local report object** A report object that is not linked to a Financial Reporting report object in Explorer. *Contrast with linked reporting object (LRO)*.

**local results** A data model's query results. Results can be used in local joins by dragging them into the data model. Local results are displayed in the catalog when requested.

**locale** A computer setting that specifies a location's language, currency and date formatting, data sort order, and the character set encoding used on the computer. Essbase uses only the encoding portion. *See also encoding* and *ESSLANG*.

**locale header record** A text record at the beginning of some non-Unicode-encoded text files, such as scripts, that identifies the encoding locale.

**location alias** A descriptor that identifies a data source. The location alias specifies a server, application, database, user name, and password. Location aliases are set by DBAs at the database level using Administration Services Console, ESSCMD, or the API.

**locked** A user-invoked process that prevents users and processes from modifying data.

**locked data model** Data models that cannot be modified by a user.

**LOCKED status** A consolidation status indicating that an entity contains data that cannot be modified.

**Log Analyzer** An Administration Services feature that enables filtering, searching, and analysis of Essbase logs.

**logic group** In FDM, contains one or more logic accounts that are generated after a source file is loaded into FDM. Logic accounts are calculated accounts that are derived from the source data.

**LRO** *See linked reporting object (LRO).*

**managed server** An application server process running in its own Java Virtual Machine (JVM).

**manual stage** A stage that requires human intervention to complete.

**Map File** Used to store the definition for sending data to or retrieving data from an external database. Map files have different extensions (.mps to send data; .mpr to retrieve data).

**Map Navigator** A feature that displays your current position on a Strategy, Accountability, or Cause and Effect map, indicated by a red outline.

**Marginal Tax Rate** Used to calculate the after-tax cost of debt. Represents the tax rate applied to the last earned income dollar (the rate from the highest tax bracket into which income falls) and includes federal, state and local taxes. Based on current level of taxable income and tax bracket, you can predict marginal tax rate.

**Market Risk Premium** The additional rate of return paid over the risk-free rate to persuade investors to hold "riskier" investments than government securities. Calculated by subtracting the risk-free rate from the expected market return. These figures should closely model future market conditions.

**master data model** An independent data model that is referenced as a source by multiple queries. When used, "Locked Data Model" is displayed in the Query section's Content pane; the data model is linked to the master data model displayed in the Data Model section, which an administrator may hide.

**mathematical operator** A symbol that defines how data is calculated in formulas and outlines. Can be any of the standard mathematical or Boolean operators; for example, +, -, *, /, and %.

**MaxL** The multidimensional database access language for Essbase, consisting of a data definition language (MaxL DDL) and a data manipulation language (MaxL DML). *See also MaxL DDL*, *MaxL DML*, and *MaxL Shell*.

**MaxL DDL** Data definition language used by Essbase for batch or interactive system-administration tasks.

**MaxL DML** Data manipulation language used in Essbase for data query and extraction.

**MaxL Perl Module** A Perl module (essbase.pm) that is part of Essbase MaxL DDL. This module can be added to the Perl package to provide access to Essbase databases from Perl programs.

**MaxL Script Editor** A script-development environment in Administration Services Console. MaxL Script Editor is an alternative to using a text editor and the MaxL Shell for administering Essbase with MaxL scripts.

**MaxL Shell** An interface for passing MaxL statements to Essbase Server. The MaxL Shell executable file is located in the Essbase bin directory (UNIX: essmsh, Windows: essmsh.exe).

**MDX (multidimensional expression)** The language that give instructions to OLE DB for OLAP- compliant databases, as SQL is used for relational databases. When you build the OLAPQuery section's Outliner, Interactive Reporting Clients translate requests into MDX instructions. When you process the query, MDX is sent to the database server, which returns records that answer your query. *See also SQL spreadsheet.*

**measures** Numeric values in an OLAP database cube that are available for analysis. Measures are margin, cost of goods sold, unit sales, budget amount, and so on. *See also fact table.*

**member** A discrete component within a dimension. A member identifies and differentiates the organization of similar units. For example, a time dimension might include such members as Jan, Feb, and Qtr1.

**member list** A named group, system- or user-defined, that references members, functions, or member lists within a dimension.

**member load** In Integration Services, the process of adding dimensions and members (without data) to Essbase outlines.

**member selection report command** A type of Report Writer command that selects member ranges based on outline relationships, such as sibling, generation, and level.

**member-specific report command** A type of Report Writer formatting command that is executed as it is encountered in a report script. The command affects only its associated member and executes the format command before processing the member.

**merge** A data load option that clears values only from the accounts specified in the data load file and replaces them with values in the data load file.

**metadata** A set of data that defines and describes the properties and attributes of the data stored in a database or used by an application. Examples of metadata are dimension names, member names, properties, time periods, and security.

**metadata elements** Metadata derived from data sources and other metadata that is stored and cataloged for Essbase Studio use.

**metadata sampling** The process of retrieving a sample of members in a dimension in a drill-down operation.

**metadata security** Security set at the member level to restrict users from accessing certain outline members.

**metaoutline** In Integration Services, a template containing the structure and rules for creating an Essbase outline from an OLAP model.

**metric** A numeric measurement computed from business data to help assess business performance and analyze company trends.

**migration** The process of copying applications, artifacts, or users from one environment or computer to another; for example, from a testing environment to a production environment.

**migration audit report** A report generated from the migration log that provides tracking information for an application migration.

**migration definition file (.mdf)** A file that contains migration parameters for an application migration, enabling batch script processing.

**migration log** A log file that captures all application migration actions and messages.

**migration snapshot** A snapshot of an application migration that is captured in the migration log.

**MIME Type** (Multipurpose Internet Mail Extension) An attribute that describes the data format of an item, so that the system knows which application should open the object. A file's mime type is determined by the file extension or HTTP header. Plug-ins tell browsers what mime types they support and what file extensions correspond to each mime type.

**mining attribute** In data mining, a class of values used as a factor in analysis of a set of data.

**minireport** A report component that includes layout, content, hyperlinks, and the query or queries to load the report. Each report can include one or more minireports.

**minischema** A graphical representation of a subset of tables from a data source that represents a data modeling context.

**missing data (#MISSING)** A marker indicating that data in the labeled location does not exist, contains no value, or was never entered or loaded. For example, missing data exists when an account contains data for a previous or future period but not for the current period.

**model** (1) In data mining, a collection of an algorithm's findings about examined data. A model can be applied against a wider data set to generate useful information about that data. (2) A file or content string containing an application-specific representation of data. Models are the basic data managed by Shared Services, of two major types: dimensional and non-dimensional application objects. (3) In Business Modeling, a network of boxes connected to represent and calculate the operational and financial flow through the area being examined.

**monetary** A money-related value.

**multidimensional database** A method of organizing, storing, and referencing data through three or more dimensions. An individual value is the intersection point for a set of dimensions. *Contrast with relational database.*

**multiload** An FDM feature that allows the simultaneous loading of multiple periods, categories, and locations.

**My Workspace Page** A page created with content from multiple sources including documents, URL, and other content types. Enables a user to aggregate content from Oracle and non-Oracle sources.

**named set** In MaxL DML, a set with its logic defined in the optional WITH section of a MaxL DML query. The named set can be referenced multiple times in the query.

**native authentication** The process of authenticating a user name and password from within the server or application.

**nested column headings** A report column heading format that displays data from multiple dimensions. For example, a column heading that contains Year and Scenario members is a nested column. The nested column heading shows Q1 (from the Year dimension) in the top line of the heading, qualified by Actual and Budget (from the Scenario dimension) in the bottom line of the heading.

**NO DATA status** A consolidation status indicating that this entity contains no data for the specified period and account.

**non-dimensional model** A Shared Services model type that includes application objects such as security files, member lists, calculation scripts, and Web forms.

**non-unique member name** *See duplicate member name.*

**note** Additional information associated with a box, measure, scorecard or map element.

**Notifications gadget** Shows notification message history received from other users or systems.

**null value** A value that is absent of data. Null values are not equal to zero.

**numeric attribute range** A feature used to associate a base dimension member that has a discrete numeric value with an attribute that represents a value range. For example, to classify customers by age, an Age Group attribute dimension can contain members for the following age ranges: 0-20, 21-40, 41-60, and 61-80. Each Customer dimension member can be associated with an Age Group range. Data can be retrieved based on the age ranges rather than on individual age values.

**ODBC** Open Database Connectivity. A database access method used from any application regardless of how the database management system (DBMS) processes the information.

**OK status** A consolidation status indicating that an entity has already been consolidated, and that data has not changed below it in the organization structure.

**OLAP Metadata Catalog** In Integration Services, a relational database containing metadata describing the nature, source, location, and type of data that is pulled from the relational data source.

**OLAP model** In Integration Services, a logical model (star schema) that is created from tables and columns in a relational database. The OLAP model is then used to generate the structure of a multidimensional database.

**online analytical processing (OLAP)** A multidimensional, multiuser, client-server computing environment for users who analyze consolidated enterprise data in real time. OLAP systems feature drill-down, data pivoting, complex calculations, trend analysis, and modeling.

**Open Database Connectivity (ODBC)** Standardized application programming interface (API) technology that allows applications to access multiple third-party databases.

**organization** An entity hierarchy that defines each entity and their relationship to others in the hierarchy.

**origin** The intersection of two axes.

**outline** The database structure of a multidimensional database, including all dimensions, members, tags, types, consolidations, and mathematical relationships. Data is stored in the database according to the structure defined in the outline.

**outline synchronization** For partitioned databases, the process of propagating outline changes from one database to another database.

**P&L accounts (P&L)** Profit and loss accounts. Refers to a typical grouping of expense and income accounts that comprise a company's income statement.

**page** A display of information in a grid or table often represented by the Z-axis. A page can contain data from one field, derived data from a calculation, or text.

**page file** Essbase data file.

**page heading** A report heading type that lists members represented on the current page of the report. All data values on the page have the members in the page heading as a common attribute.

**page member** A member that determines the page axis.

**palette** A JASC compliant file with a .PAL extension. Each palette contains 16 colors that complement each other and can be used to set the dashboard color elements.

**parallel calculation** A calculation option. Essbase divides a calculation into tasks and calculates some tasks simultaneously.

**parallel data load** In Essbase, the concurrent execution of data load stages by multiple process threads.

**parallel export** The ability to export Essbase data to multiple files. This may be faster than exporting to a single file, and it may resolve problems caused by a single data file becoming too large for the operating system to handle.

**parent adjustments** The journal entries that are posted to a child in relation to its parent.

**parents** The entities that contain one or more dependent entities that report directly to them. Because parents are both entities and associated with at least one node, they have entity, node, and parent information associated with them.

**partition area** A sub cube within a database. A partition is composed of one or more areas of cells from a portion of the database. For replicated and transparent partitions, the number of cells within an area must be the same for the data source and target to ensure that the two partitions have the same shape. If the data source area contains 18 cells, the data target area must also contain 18 cells to accommodate the number of values.

**partitioning** The process of defining areas of data that are shared or linked between data models. Partitioning can affect the performance and scalability of Essbase applications.

**pattern matching** The ability to match a value with any or all characters of an item entered as a criterion. Missing characters may be represented by wild card values such as a question mark (?) or an asterisk (*). For example, “Find all instances of apple” returns apple, but “Find all instances of apple*” returns apple, applesauce, applecranberry, and so on.

**percent consolidation** The portion of a child's values that is consolidated to its parent.

**percent control** Identifies the extent to which an entity is controlled within the context of its group.

**percent ownership** Identifies the extent to which an entity is owned by its parent.

**performance indicator** An image file used to represent measure and scorecard performance based on a range you specify; also called a status symbol. You can use the default performance indicators or create an unlimited number of your own.

**periodic value method (PVA)** A process of currency conversion that applies the periodic exchange rate values over time to derive converted results.

**permission** A level of access granted to users and groups for managing data or other users and groups.

**persistence** The continuance or longevity of effect for any Essbase operation or setting. For example, an Essbase administrator may limit the persistence of user name and password validity.

**personal pages** A personal window to repository information. You select what information to display and its layout and colors.

**personal recurring time events** Reusable time events that are accessible only to the user who created them.

**personal variable** A named selection statement of complex member selections.

**perspective** A category used to group measures on a scorecard or strategic objectives within an application. A perspective can represent a key stakeholder (such as a customer, employee, or shareholder/financial) or a key competency area (such as time, cost, or quality).

**pie chart** A chart that shows one data set segmented in a pie formation.

**pinboard** One of the three data object display types. Pinboards are graphics, composed of backgrounds and interactive icons called pins. Pinboards require traffic lighting definitions.

**pins** Interactive icons placed on graphic reports called pinboards. Pins are dynamic. They can change images and traffic lighting color based on the underlying data values and analysis tools criteria.

**pivot** The ability to alter the perspective of retrieved data. When Essbase first retrieves a dimension, it expands data into rows. You can then pivot or rearrange the data to obtain a different viewpoint.

**planner** Planners, who comprise the majority of users, can input and submit data, use reports that others create, execute business rules, use task lists, enable e-mail notification for themselves, and use Smart View.

**planning unit** A data slice at the intersection of a scenario, version, and entity; the basic unit for preparing, reviewing, annotating, and approving plan data.

**plot area** The area bounded by X, Y, and Z axes; for pie charts, the rectangular area surrounding the pie.

**plug account** An account in which the system stores any out of balance differences between intercompany account pairs during the elimination process.

**post stage assignment** Assignments in the allocation model that are assigned to locations in a subsequent model stage.

**POV (point of view)** A feature for setting data focus by selecting members that are not already assigned to row, column, or page axes. For example, selectable POVs in FDM could include location, period, category, and target category. In another example, using POV as a filter in Smart View, you could assign the Currency dimension to the POV and select the Euro member. Selecting this POV in data forms displays data in Euro values.

**precalculation** Calculating the database prior to user retrieval.

**precision** Number of decimal places displayed in numbers.

**predefined drill paths** Paths used to drill to the next level of detail, as defined in the data model.

**presentation** A playlist of Web Analysis documents, enabling reports to be grouped, organized, ordered, distributed, and reviewed. Includes pointers referencing reports in the repository.

**preserve formulas** User-created formulas kept within a worksheet while retrieving data.

**primary measure** A high-priority measure important to your company and business needs. Displayed in the Contents frame.

**process monitor report** Displays a list of locations and their positions within the FDM data conversion process. You can use the process monitor report to monitor the status of the closing process. The report is time-stamped. Therefore, it can be used to determine to which locations at which time data was loaded.

**product** In Shared Services, an application type, such as Planning or Performance Scorecard.

**Production Reporting** *See SQR Production Reporting*.

**project** An instance of EPM System products grouped together in an implementation. For example, a Planning project may consist of a Planning application, an Essbase cube, and a Financial Reporting server instance.

**property** A characteristic of an artifact, such as size, type, or processing instructions.

**provisioning** The process of granting users and groups specific access permissions to resources.

**proxy server** A server acting as an intermediary between workstation users and the Internet to ensure security.

**public job parameters** Reusable, named job parameters created by administrators and accessible to users with requisite access privileges.

**public recurring time events** Reusable time events created by administrators and accessible through the access control system.

**PVA** *See periodic value method (PVA)*.

**qualified name** A member name in a qualified format that differentiates duplicate member names in a duplicate member outline. For example, [Market].[East].[State].[New York] or [Market].[East].[City].[New York]

**query** Information requests from data providers. For example, used to access relational data sources.

**query governor** An Essbase Integration server parameter or Essbase server configuration setting that controls the duration and size of queries made to data sources.

**range** A set of values including upper and lower limits, and values falling between limits. Can contain numbers, amounts, or dates.

**reciprocal assignment** An assignment in the financial flow that also has the source as one of its destinations.

**reconfigure URL** URL used to reload servlet configuration settings dynamically when users are already logged on to the Workspace.

**record** In a database, a group of fields making up one complete entry. For example, a customer record may contain fields for name, address, telephone number, and sales data.

**recurring template** A journal template for making identical adjustments in every period.

**recurring time event** An event specifying a starting point and the frequency for running a job.

**redundant data** Duplicate data blocks that Essbase retains during transactions until Essbase commits updated blocks.

**regular journal** A feature for entering one-time adjustments for a period. Can be balanced, balanced by entity, or unbalanced.

**Related Accounts** The account structure groups all main and related accounts under the same main account number. The main account is distinguished from related accounts by the first suffix of the account number.

**relational database** A type of database that stores data in related two-dimensional tables. *Contrast with multidimensional database*.

**replace** A data load option that clears existing values from all accounts for periods specified in the data load file, and loads values from the data load file. If an account is not specified in the load file, its values for the specified periods are cleared.

**replicated partition** A portion of a database, defined through Partition Manager, used to propagate an update to data mastered at one site to a copy of data stored at another site. Users can access the data as though it were part of their local database.

**Report Extractor** An Essbase component that retrieves report data from the Essbase database when report scripts are run.

**report object** In report designs, a basic element with properties defining behavior or appearance, such as text boxes, grids, images, and charts.

**report script** A text file containing Essbase Report Writer commands that generate one or more production reports.

**Report Viewer** An Essbase component that displays complete reports after report scripts are run.

**reporting currency** The currency used to prepare financial statements, and converted from local currencies to reporting currencies.

**repository** Stores metadata, formatting, and annotation information for views and queries.

**resources** Objects or services managed by the system, such as roles, users, groups, files, and jobs.

**restore** An operation to reload data and structural information after a database has been damaged or destroyed, typically performed after shutting down and restarting the database.

**restructure** An operation to regenerate or rebuild the database index and, in some cases, data files.

**result frequency** The algorithm used to create a set of dates to collect and display results.

**review level** A Process Management review status indicator representing the process unit level, such as Not Started, First Pass, Submitted, Approved, and Published.

**Risk Free Rate** The rate of return expected from "safer" investments such as long-term U.S. government securities.

**role** The means by which access permissions are granted to users and groups for resources.

**roll-up** *See consolidation.*

**root member** The highest member in a dimension branch.

**RSC services** Services that are configured with Remote Service Configurator, including Repository Service, Service Broker, Name Service, Event Service, and Job Service.

**runtime prompt** A variable that users enter or select before a business rule is run.

**sampling** The process of selecting a representative portion of an entity to determine the entity's characteristics. *See also metadata sampling.*

**saved assumptions** User-defined Planning assumptions that drive key business calculations (for example, the cost per square foot of office floor space).

**scaling** Scaling determines the display of values in whole numbers, tens, hundreds, thousands, millions, and so on.

**scenario** A dimension for classifying data (for example, Actuals, Budget, Forecast1, and Forecast2).

**scope** The area of data encompassed by any Essbase operation or setting; for example, the area of data affected by a security setting. Most commonly, scope refers to three levels of granularity, where higher levels encompass lower levels. From highest to lowest, these levels are as follows: the entire system (Essbase Server), applications on Essbase servers, or databases within Essbase server applications. *See also persistence.*

**score** The level at which targets are achieved, usually expressed as a percentage of the target.

**scorecard** Business object that represents the progress of an employee, strategy element, or accountability element toward goals. Scorecards ascertain this progress based on data collected for each measure and child scorecard added to the scorecard.

**scraping** An inspection of a data source to derive the most basic metadata elements from it. *Contrast with introspection.*

**Search gadget** Searches the Reporting and Analysis repository. The Search gadget looks for a match in the document keywords and description, which are set when you import a document.

**secondary measure** A low-priority measure, less important than primary measures. Secondary measures do not have Performance reports but can be used on scorecards and to create dimension measure templates.

**security agent** A Web access management provider (for example, Netegrity SiteMinder) that protects corporate Web resources.

**security platform** A framework enabling EPM System products to use external authentication and single sign-on.

**serial calculation** The default calculation setting. Divides a calculation pass into tasks and calculates one task at a time.

**services** Resources that enable business items to be retrieved, changed, added, or deleted. Examples: Authorization and Authentication.

**servlet** A piece of compiled code executable by a Web server.

**shared member** A member that shares storage space with another member of the same name, preventing duplicate calculation of members that occur multiple times in an Essbase outline.

**Shared Services Registry** Part of the Shared Services database, the Shared Services Registry stores and re-uses information for most installed EPM System products, including installation directories, database settings, deployment settings, computer names, ports, servers, URLs, and dependent service data.

**Shared Workspace Page** Workspace Pages shared across an organization which are stored in a special System folder and can be accessed by authorized users from the Shared Workspace Pages Navigate menu.

**sibling** A child member at the same generation as another child member and having the same immediate parent. For example, the members Florida and New York are children of East and each other's siblings.

**single sign-on** Ability to access multiple EPM System products after a single login using external credentials.

**smart slice** In Smart View, a reusable perspective of a data source that contains a restricted set of dimensions or dimension members.

**Smart Space client software** Runs on the client's computer and provides gadgets, instant collaboration and access to the Reporting and Analysis repository. It is composed of the Smart Space framework and gadgets.

**Smart Space Collaborator** A service that enables users or systems to send messages and share Reporting and Analysis repository content. The message can take many forms, including instant message style discussions, meetings, and toast messages.

**smart tags** Keywords in Microsoft Office applications that are associated with predefined actions available from the Smart Tag menu. In EPM System products, smart tags can also be used to import Reporting and Analysis content, and access Financial Management and Essbase functions.

**SmartBook gadget** Contains documents from the Reporting and Analysis repository or URLs. All documents are loaded when the SmartBook is opened so you can access all content immediately.

**SmartCut** A link to a repository item, in URL form.

**snapshot** Read-only data from a specific time.

**source currency** The currency from which values originate and are converted through exchange rates to the destination currency.

**sparse dimension** In block storage databases, a dimension unlikely to contain data for all member combinations when compared to other dimensions. For example, not all customers have data for all products. *Contrast with dense dimension*.

**SPF files** Printer-independent files created by an SQR Production Reporting server, containing a representation of the actual formatted report output, including fonts, spacing, headers, footers, and so on.

**Spotlighter** A tool that enables color coding based on selected conditions.

**SQL spreadsheet** A data object that displays the result set of a SQL query.

**SQR Production Reporting** A specialized programming language for data access, data manipulation, and creating SQR Production Reporting documents.

**stage** A task description that forms one logical step within a taskflow, usually performed by an individual. A stage can be manual or automated.

**stage action** For automated stages, the invoked action that executes the stage.

**staging area** A database that you create to meet the needs of a specific application. A staging area is a snapshot or restructured version of one or more RDBMSs.

**standard dimension** A dimension that is not an attribute dimension.

**standard journal template** A journal function used to post adjustments that have common adjustment information for each period. For example, you can create a standard template that contains the common account IDs, entity IDs, or amounts, then use the template as the basis for many regular journals.

**Status bar** The status bar at the bottom of the screen displays helpful information about commands, accounts, and the current status of your data file.

**stored hierarchy** In aggregate storage databases outlines only. A hierarchy in which the members are aggregated according to the outline structure. Stored hierarchy members have certain restrictions, for example, they cannot contain formulas.

**strategic objective (SO)** A long-term goal defined by measurable results. Each strategic objective is associated with one perspective in the application, has one parent, the entity, and is a parent to critical success factors or other strategic objectives.

**Strategy map** Represents how the organization implements high-level mission and vision statements into lower-level, constituent strategic goals and objectives.

**structure view** Displays a topic as a simple list of component data items.

**Structured Query Language** A language used to process instructions to relational databases.

**Subaccount Numbering** A system for numbering subaccounts using non-sequential, whole numbers.

**subscribe** Flags an item or folder to receive automatic notification whenever the item or folder is updated.

**Summary chart** In the Investigates Section, rolls up detail charts shown below in the same column, plotting metrics at the summary level at the top of each chart column.

**super service** A special service used by the startCommonServices script to start the RSC services.

**supervisor** A user with full access to all applications, databases, related files, and security mechanisms for a server.

**supporting detail** Calculations and assumptions from which the values of cells are derived.

**suppress rows** Excludes rows containing missing values, and underscores characters from spreadsheet reports.

**symmetric multiprocessing (SMP)** A server architecture that enables multiprocessing and multithreading. Performance is not significantly degraded when a large number of users connect to an single instance simultaneously.

**sync** Synchronizes Shared Services and application models.

**synchronized** The condition that exists when the latest version of a model resides in both the application and in Shared Services. *See also model*.

**system extract** Transfers data from an application's metadata into an ASCII file.

**tabs** Navigable views of accounts and reports in Strategic Finance.

**target** Expected results of a measure for a specified period of time (day, quarter, and so on).

**task list** A detailed status list of tasks for a particular user.

**taskflow** The automation of a business process in which tasks are passed from one taskflow participant to another according to procedural rules.

**taskflow definition** Represents business processes in the taskflow management system. Consists of a network of stages and their relationships; criteria indicating the start and end of the taskflow; and information about individual stages, such as participants, associated applications, associated activities, and so on.

**taskflow instance** Represents a single instance of a taskflow including its state and associated data.

**taskflow management system** Defines, creates, and manages the execution of a taskflow including: definitions, user or application interactions, and application executables.

**taskflow participant** The resource who performs the task associated with the taskflow stage instance for both manual and automated stages.

**Taxes - Initial Balances** Strategic Finance assumes that the Initial Loss Balance, Initial Gain Balance and the Initial Balance of Taxes Paid entries have taken place in the period before the first Strategic Finance time period.

**TCP/IP** *See Transmission Control Protocol/Internet Protocol (TCP/IP)*.

**template** A predefined format designed to retrieve particular data consistently.

**text list** In Essbase, an object that stores text values mapped to numeric identifiers. Text Lists enable the use of text measures.

**text measure** A data type that allows measure values to be expressed as text. In Essbase, a member tagged as “Text” in the dimension where measures are represented. The cell values are displayed as predefined text. For example, the text measure "Satisfaction Index" may have the values Low, Medium, and High. *See also typed measure*, *text list*, *derived text measure*.

**time dimension** Defines the time period that the data represents, such as fiscal or calendar periods.

**time events** Triggers for execution of jobs.

**time line viewer** An FDM feature that allows a user to view dates and times of completed process flow steps for specific locations.

**time scale** Displays metrics by a specific period in time, such as monthly or quarterly.

**time series reporting** A process for reporting data based on a calendar date (for example, year, quarter, month, or week).

**Title bar** Displays the Strategic Finance name, the file name, and the scenario name Version box.

**toast message** Messages that appear in the lower right corner of the screen and fade in and out.

**token** An encrypted identification of one valid user or group on an external authentication system.

**top and side labels** Column and row headings on the top and sides of a Pivot report.

**top-level member** A dimension member at the top of the tree in a dimension outline hierarchy, or the first member of the dimension in sort order if there is no hierarchical relationship among dimension members. The top-level member name is generally the same as the dimension name if a hierarchical relationship exists.

**trace allocations** A feature of Profitability and Cost Management that enables you to visually follow the flow of financial data, either forwards or backwards, from a single intersection throughout the model.

**trace level** Defines the level of detail captured in the log file.

**traceability** The ability to track a metadata element to its physical source. For example, in Essbase Studio, a cube schema can be traced from its hierarchies and measure hierarchies, to its dimension elements, date/time elements, and measures, and ultimately, to its physical source elements.

**traffic lighting** Color-coding of report cells, or pins based on a comparison of two dimension members, or on fixed limits.

**transformation** (1) Transforms artifacts so that they function properly in the destination environment after application migration. (2) In data mining, modifies data (bidirectionally) flowing between the cells in the cube and the algorithm.

**translation** *See currency conversion.*

**Transmission Control Protocol/Internet Protocol (TCP/IP)** A standard set of communication protocols linking computers with different operating systems and internal architectures. TCP/IP utilities are used to exchange files, send mail, and store data to various computers that are connected to local and wide area networks.

**transparent login** Logs in authenticated users without launching the login screen.

**transparent partition** A shared partition that enables users to access and change data in a remote database as though it is part of a local database

**triangulation** A means of converting balances from one currency to another via a third common currency. In Europe, this is the euro for member countries. For example, to convert from French franc to Italian lira, the common currency is defined as European euro. Therefore, in order to convert balances from French franc to Italian lira, balances are converted from French franc to European euro and from European euro to Italian lira.

**triggers** An Essbase feature whereby data is monitored according to user-specified criteria which when met cause Essbase to alert the user or system administrator.

**trusted password** A password that enables users authenticated for one product to access other products without reentering their passwords.

**trusted user** Authenticated user.

**tuple** MDX syntax element that references a cell as an intersection of a member from each dimension. If a dimension is omitted, its top member is implied. Examples: (Jan); (Jan, Sales); ( [Jan], [Sales], [Cola], [Texas], [Actual] )

**two-pass** An Essbase property that is used to recalculate members that are dependent on the calculated values of other members. Two-pass members are calculated during a second pass through the outline.

**typed measure** In Essbase, a member tagged as "Text" or "Date" in the dimension where measures are represented. The cell values are displayed as predefined text or dates.

**unary operator** A mathematical indicator (+, -, *, /, %) associated with an outline member. The unary operator defines how the member is calculated during a database roll-up.

**Unicode-mode application** An Essbase application wherein character text is encoded in UTF-8, enabling users with computers set up for different languages to share application data.

**Uniform Resource Locator** The address of a resource on the Internet or an intranet.

**unique member name** A non-shared member name that exists only once in a database outline.

**unique member outline** A database outline that is not enabled for duplicate member names.

**upgrade** The process of replacing an earlier software release with a current release or replacing one product with another.

**upper-level block** A type of data block wherein at least one of the sparse members is a parent-level member.

**user directory** A centralized location for user and group information. Also known as a repository or provider.

**user variable** Dynamically renders data forms based on a user's member selection, displaying only the specified entity. For example, user variable named Department displays specific departments and employees.

**user-defined attribute (UDA)** User-defined attribute, associated with members of an outline to describe a characteristic of the members. Users can use UDAs to return lists of members that have the specified UDA associated with them.

**user-defined member list** A named, static set of members within a dimension defined by the user.

**validation** A process of checking a business rule, report script, or partition definition against the outline to make sure that the object being checked is valid. For example, in FDM, validation rules ensure that certain conditions are met after data is loaded from FDM to the target application.

**value dimension** Used to define input value, translated value, and consolidation detail.

**variance** Difference between two values (for example, planned and actual value).

**varying attribute** An attribute association that changes over one or more dimensions. It can be used to track a value in relation to these dimensions; for example, the varying attribute Sales Representative, associated with the Product dimension, can be used to track the value Customer Sales of several different sales representatives in relation to the Time dimension. Varying attributes can also be used for member selection, such as finding the Products that a Sales Representative was responsible for in May.

**version** Possible outcome used within the context of a scenario of data. For example, Budget - Best Case and Budget - Worst Case where Budget is scenario and Best Case and Worst Case are versions.

**view** Representation of either a year-to-date or periodic display of data.

**visual cue** A formatted style, such as a font or a color, that highlights specific types of data values. Data values may be dimension members; parent, child, or shared members; dynamic calculations; members containing a formula; read only data cells; read and write data cells; or linked objects.

**Web server** Software or hardware hosting intranet or Internet Web pages or Web applications.

**weight** Value assigned to an item on a scorecard that indicates the relative importance of that item in the calculation of the overall scorecard score. The weighting of all items on a scorecard accumulates to 100%. For example, to recognize the importance of developing new features for a product, the measure for New Features Coded on a developer's scorecard would be assigned a higher weighting than a measure for Number of Minor Defect Fixes.

**wild card** Character that represents any single character or group of characters (*) in a search string.

**WITH section** In MaxL DML, an optional section of the query used for creating re-usable logic to define sets or members. Sets or custom members can be defined once in the WITH section, and then referenced multiple times during a query.

**work flow** The steps required to process data from start to finish in FDM. The workflow consists of Import (loading data from the GL file), Validate (ensures all members are mapped to a valid account), Export (loads the mapped members to the target application), and Check (verifies accuracy of data by processing data with user-defined validation rules).

**workbook** An entire spreadsheet file with many worksheets.

**Workspace Page** A page created with content from multiple sources including documents, URL, and other content types. Enables a user to aggregate content from Oracle and non-Oracle sources.

**write-back** The ability for a retrieval client, such as a spreadsheet, to update a database value.

**ws.conf** A configuration file for Windows platforms.

**wsconf_platform** A configuration file for UNIX platforms.

**XML** *See Extensible Markup Language (XML).*

**XOLAP** An Essbase multidimensional database that stores only the outline metadata and retrieves all data from a relational database at query time. XOLAP supports aggregate storage databases and applications that contain duplicate member names.

**Y axis scale** Range of values on Y axis of charts displayed in Investigate Section. For example, use a unique Y axis scale for each chart, the same Y axis scale for all Detail charts, or the same Y axis scale for all charts in the column. Often, using a common Y axis improves your ability to compare charts at a glance.

**Zero Administration** Software tool that identifies version number of the most up-to-date plug-in on the server.

**zoom** Sets the magnification of a report. For example, magnify a report to fit whole page, page width, or percentage of magnification based on 100%.

**ZoomChart** Used to view detailed information by enlarging a chart. Enables you to see detailed numeric information on the metric that is displayed in the chart.

# Index

## D

## P



## Q



## R



## S



## T

## U



## V



## W