Oracle® Hyperion Profitability and Cost Management

Administrator's Guide

Release 11.1.2.4.110
# Contents

**Part I. Introduction to Profitability and Cost Management** ............................................... 15

**Chapter 1. Overview** ................................................................ ........................................ 17
  - Accessibility in Profitability and Cost Management .................................................. 18
  - Profitability Types ................................................................................................. 18
    - Standard Profitability ....................................................................................... 18
    - Detailed Profitability ....................................................................................... 19
    - Management Ledger Profitability .................................................................. 19
  - Additional Product Components ........................................................................... 20
  - Administrative Tasks .......................................................................................... 20
  - Launching Profitability and Cost Management ................................................... 21

**Chapter 2. Managing Security and Authorizing Users** ............................................ 23
  - About User Setup and Provisioning ....................................................................... 23
  - Assigning Security Roles ...................................................................................... 24
    - Security Roles for Standard Profitability ....................................................... 26
    - Security Roles for Detailed Profitability ....................................................... 28
    - Security Roles for Management Ledger Profitability ..................................... 31
  - Auditing Changes in Profitability and Cost Management ..................................... 32
  - Output Log Files .................................................................................................. 35
  - Oracle Diagnostic Logging (ODL) Files ............................................................. 36

**Part II. Working with Standard Profitability Applications** ........................................ 37

**Chapter 3. About Standard Profitability** ................................................................. 39
  - Understanding the Standard Profitability Product Architecture ................................ 39
  - Working with Databases for Standard Profitability ............................................. 40

**Chapter 4. Working with Standard Profitability and Cost Management Dimensions and Members** ................. 43
  - About Standard Profitability Dimensions and Metadata ......................................... 43
Creating Management Ledger Profitability Applications .............. 188
Adding Management Ledger Profitability Dimensions Using the Wizard ...... 189
Validating and Deploying Management Ledger Applications Using the Wizard .... 191

Part V. Appendices ........................................................................................................ 193

Appendix A. Creating the First Profitability and Cost Management Application with Performance Management Architect ........................................ 195
About Creating the First Application with Performance Management Architect ...... 195
Steps to Create the First Application with Performance Management Architect ...... 195

Appendix B. Creating Applications Using the Profitability Applications Console ........................................ 199
Guidelines for Using the Profitability Applications Console ..................... 199
Displaying the Profitability Applications Console ............................ 200
Creating Profitability and Cost Management Applications in the Profitability Applications Console .......................................................... 201
Creating Applications with Dimensions from an Essbase Master Cube ...... 201
Creating Management Ledger Applications with Dimensions from Flat Files .... 203
Creating Management Ledger Applications by Importing Template Files ...... 203
Working with Applications in the Profitability Applications Console .......... 204
Editing Application Descriptions and Shared Services Projects ............... 205
Performing Other Application Actions .............................................. 205
Updating Application Dimensions ...................................................... 206
Viewing Tasks in the Profitability Applications Console Job Library .......... 207
Preparing Templates and Flat Files for Creating and Updating Management Ledger Profitability Applications ....................................................... 210
Exporting Templates ........................................................................... 210
Preparing Flat Files for Each Management Ledger Dimension ............... 211
About Flat File Properties ................................................................. 212
About Comments in Flat Files .............................................................. 213
Flat File Sample .................................................................................. 214
Creating an Essbase Master Cube for Profitability and Cost Management .... 214
Dimension Types .................................................................................. 215
User-Defined Attributes ................................................................. 216
ASO Dimension and Member Properties ........................................... 217
Specifying BSO and Member Dimension Properties ............................ 218
Viewing Block Storage Option (BSO) Data Storage Values Assigned to Member Properties .............................................................. 221
Custom Measures Dimensions .............................................................. 221

Appendix C. Importing Data into Profitability and Cost Management ................ 223
About Staging Tables .............................................................................. 224
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creating Import Configurations</td>
<td>225</td>
</tr>
<tr>
<td>Modifying Import Configurations</td>
<td>227</td>
</tr>
<tr>
<td>Deleting Import Configurations</td>
<td>228</td>
</tr>
<tr>
<td>Running Import Configurations</td>
<td>228</td>
</tr>
<tr>
<td>Verifying Imported Data</td>
<td>229</td>
</tr>
<tr>
<td><strong>Appendix D. Migrating Data Using EPM System Lifecycle Management</strong></td>
<td></td>
</tr>
<tr>
<td>About Lifecycle Management</td>
<td>231</td>
</tr>
<tr>
<td>Modifying Default Timeout Settings for Lifecycle Management</td>
<td>232</td>
</tr>
<tr>
<td><strong>Appendix E. Backing Up Profitability and Cost Management Components</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Appendix F. Essbase Naming Conventions</strong></td>
<td></td>
</tr>
<tr>
<td>Generated Calculation Script Naming Conventions</td>
<td>237</td>
</tr>
<tr>
<td>Essbase Naming Restrictions for Applications and Databases</td>
<td>238</td>
</tr>
<tr>
<td>Essbase Naming Restrictions for Dimensions, Members, and Aliases</td>
<td>239</td>
</tr>
<tr>
<td>Essbase Naming Conventions for Attribute Calculations Dimension</td>
<td>242</td>
</tr>
</tbody>
</table>
Documentation Accessibility

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

Access to Oracle Support

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

Follow EPM Information Development on these social media sites:

LinkedIn - http://www.linkedin.com/groups?gid=3127051&goback=.gmp_3127051
Twitter - http://twitter.com/hyperionepminfo
Facebook - http://www.facebook.com/pages/Hyperion-EPM-Info/102682103112642
Google+ - https://plus.google.com/106915048672979407731/#106915048672979407731/posts
YouTube - http://www.youtube.com/user/OracleEPMWebcasts
In Introduction to Profitability and Cost Management:

- Overview
- Managing Security and Authorizing Users
Oracle Hyperion Profitability and Cost Management is an analytical application that is accessed from Oracle Hyperion Enterprise Performance Management Workspace, and is 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. For Profitability application types, see “Profitability Types” on page 18.
- Use Oracle 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 and Profitability Applications 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.

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.
This guide assumes that you will use Performance Management Architect for managing your applications and dimensions for your models. However, instead of using Performance Management Architect you can use Profitability Applications instead (Appendix B, “Creating Applications Using the Profitability Applications Console”).

- Use Oracle Essbase or a relational database to create the application outline and to store and execute calculation scripts.
- 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 Hyperion Reporting and Analysis, Oracle Hyperion Financial Reporting, Oracle 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 *Oracle Hyperion Profitability and Cost Management Accessibility Guide*. These features are unique to Profitability and Cost Management.

**Profitability Types**

Profitability and Cost Management offers two different types of applications that are used to evaluate profitability:

- “Standard Profitability” on page 18
- “Detailed Profitability” on page 19
- “Management Ledger Profitability” on page 19

**Standard Profitability**

Standard Profitability focuses on contribution analysis, following the flow of cost and revenue funds through all stages of the process to determine where funds are coming from, and where they are going.

A Standard Profitability and Cost Management model enables you to monitor and control direct contribution data for your entire model. The input amounts, the flow of cost and revenue, and the final destination of the funds can be tracked for both cost and revenue to ensure the resources are used to best advantage, and profitability can be easily demonstrated. Calculation results are posted to individual cost centers or accounts.

Data for the Standard Profitability model is housed in both Essbase multidimensional databases and relational databases. You create the model in Performance Management Architect, and define the hierarchy of accounts, activities and operations within your organization using
dimensions and dimension members. An AllocationType dimension is imported from Performance Management Architect. This dimension is used to correctly allocate costs and revenue, and store direct allocations and allocation genealogy.

For more information on working with Standard Profitability, see Chapter 3, “About Standard Profitability”.

**Detailed Profitability**

Detailed Profitability provides a single-step allocation of pools or rates from a single source to a destination for the purpose of analyzing profitability. Detailed Profitability utilizes a relational database for model artifact storage, calculations, and reporting views.

A Detailed Profitability and Cost Management model employs a user-defined schema to organize relational tables with existing data and associated lookup table to extend that data. Data for the Detailed Profitability model is housed in relational databases only.

You create the model in Performance Management Architect, and define the business dimensions, aliases, measures, and so on within your organization. Within Profitability and Cost Management, the data is mapped to the application to enable you to build the Detailed Profitability model. The application can handle extremely large volumes.

The application does not use a hierarchical structure, but processes all allocation in a flow between a single Source and Destination combination. Allocation is handled through a restricted MeasuresDetailed dimension, rather than creating an AllocationType as in Standard Profitability. The MembersDetailed dimension contains a limited set of members to process all allocations.

For more information on working with Detailed Profitability, see Chapter 8, “Understanding the Detailed Profitability Product Architecture”.

**Management Ledger Profitability**

Management Ledger applications are designed for use by analysts who have deep domain experience in the computation and reporting methods of management reporting, but who may not have much experience with Essbase and scripting syntax or programming languages.

Similar to Standard Profitability applications, data for Management Ledger applications is housed in both Essbase multidimensional databases and relational databases. You create applications in Performance Management Architect, and define the hierarchy of accounts, activities, and operations within the organization using dimensions and dimension members.

After the application is deployed, you build the model to show the flow of funds to specific cost and revenue allocations. Both the source and destination ranges of allocations are defined as calculation rules using the Profitability and Cost Management user interface. As for Standard and Detailed Profitability application types, points of view (POVs) represent a specific instance of the model, and can be used to view or calculate different versions of a model; for example, to view values for different months or quarters, to compare budget versus actual figures, or to create scenarios to measure the impact of various changes on the bottom line.
In Management Ledger models, there is no concept of stages or layers. All structure is controlled through the organization of rule sets and rules under POVs. For each POV, calculation rules are organized into groups that run against the same or similar region of the database and at the same or similar time. These groups are called rule sets. They determine the order in which calculation rules run. Calculation rules can inherit default member selections from the POV or rule set level so users can define a region of the database once and use it many times without having to specify it each time. These defaults are called “contexts”.

The model is validated after creation to ensure that all allocations are accounted for, and calculations are balanced. Following validation, you deploy the database, and then calculate the model, and analyze the results. For details, see the Oracle Hyperion Profitability and Cost Management User’s Guide and Part IV of this Guide, “Working with Management Ledger Applications.”

### Additional Product Components

Profitability and Cost Management is used to create models and perform allocations. The power and flexibility of Profitability and Cost Management are extended through the use of the products shown in **Table 1, “Profitability and Cost Management Product Components”**.

<table>
<thead>
<tr>
<th>Product</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Essbase</td>
<td>For Standard Profitability only, store and calculate Profitability and Cost Management application data.</td>
</tr>
<tr>
<td>Oracle Essbase Administration Services</td>
<td>For Standard Profitability only, interface to Essbase server, using the Essbase Administration Services Console. Used to design, develop, maintain, and manage multiple Essbase applications and databases.</td>
</tr>
<tr>
<td>Performance Management Architect or Profitability and Cost Management Profitability Applications feature</td>
<td>Manage dimensions and applications.</td>
</tr>
<tr>
<td>EPM Workspace</td>
<td>Navigate to Profitability and Cost Management and other products, and manage applications.</td>
</tr>
<tr>
<td>Shared Services</td>
<td>Provision users from external systems for Profitability and Cost Management. Migrate Profitability and Cost Management applications.</td>
</tr>
<tr>
<td>Oracle Smart View for Office</td>
<td>Enter and report on Profitability and Cost Management data in Microsoft Excel spreadsheets.</td>
</tr>
<tr>
<td>Financial Reporting</td>
<td>Create reports and charts for Web or print distribution.</td>
</tr>
<tr>
<td>Web Analysis</td>
<td>Use a Web client to create ad hoc reports and charts for analysis.</td>
</tr>
</tbody>
</table>

### 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.
- Manage the Shared Library and Profitability and Cost Management dimensions and members in Performance Management Architect or Profitability and Cost Management Profitability Applications feature.
- Generate multidimensional Essbase databases.
- Create, update, and delete model stages, drivers and points of view (POVs).
- Create, update, and delete driver selections, assignments, assignment rules and assignment rules' selections.
- Create, update, and delete Calculation Rules.
- Calculate and re-calculate the model.
- 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 these sections:
- Chapter 4, "Working with Standard Profitability and Cost Management Dimensions and Members"
- Chapter 12, "Working with Detailed Profitability and Cost Management Dimensions and Metadata"

**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 (If running in Linux or UNIX, Performance Management Architect is not applicable)
   - Essbase (for Standard Profitability only)
Profitability and Cost Management

See the Oracle Enterprise Performance Management System Installation and Configuration Guide and the Oracle 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/ where the server name is the Oracle HTTP Server (OHS) server name.

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 Select Navigate, then Applications, then Profitability and then select the application you want to view.
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 Enterprise Performance Management System Installation and Configuration Guide.

Two guides are available for managing security and user provisioning:

- Use the Oracle Enterprise Performance Management System User Security Administration Guide for technical information regarding security:
  - SSL (one-way, two-way, SSL OffLoading, SSL Termination)
  - Single Sign-On
  - Default EPM System SSO
  - Security Agents
  - Custom Login
  - Custom Authentication Modules
  - Guidelines for securing the EPM System

- Use the Oracle Enterprise Performance Management System User Security Administration Guide for information on how to set up and manage user provisioning:
  - Oracle Hyperion Shared Services Console
  - User Directories
  - Applications and Application Groups
  - Delegated User Management
Managing Native Directory
Managing Provisioning
Provisioning the EPM System

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 communication between applications is seamless 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 EPM Workspace, select Navigate, then Administrator, and then Shared Services Console to access the Shared Services screens.
   
   Note: The first time that you log on, an administrator (admin) user is automatically created for your product.

2. Set the user assigned to the admin role to the Provisioning Manager role. See the Oracle Enterprise Performance Management System User Security Administration Guide.

3. Create users. See the Oracle Enterprise Performance Management System User Security Administration Guide.

4. 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 tasks outside of Profitability and Cost Management, you must also provision the selected user with the EPM Workspace role, for example, Lifecycle Management.

5. Create groups, as required. You can provision groups within groups. See the Oracle Enterprise Performance Management System User Security Administration Guide.


7. 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 Enterprise Performance Management System User 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 multiple roles. Permission for a specific action, is checked at the time the action is initiated.

**Note:** At least one user must be manually assigned the Provisioning Manager role in the Shared Services Console. This Provisioning Manager role enables that user to assign security roles to other users for the application. See the *Oracle Enterprise Performance Management System User Security Administration Guide*.

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

If a user must initiate and monitor taskflows, additional Shared Services roles are required, as shown on Table 2, “Standard Profitability and Cost Management Security Roles”.

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

The security roles detailed in the following sections are specific to Profitability and Cost Management. For a complete description of all security roles, see the *Oracle Enterprise Performance Management System User Security Administration Guide* for detailed instructions.
## Security Roles for Standard Profitability

### Table 2  
Standard Profitability and Cost Management Security Roles

<table>
<thead>
<tr>
<th>Security Role</th>
<th>Type of Role</th>
<th>Description</th>
</tr>
</thead>
</table>
| Administrator     | Power        | - Create and maintain user accounts and security roles, and provision users, using Shared Services  
|                   |              | - Generate Essbase databases  
| (admin)           |              | - Set up and maintain application preferences  
|                   |              | - Build the model database using Performance Management Architect to select the common dimensions and members  
|                   |              | - Create and maintain elements within the model, such as stages, drivers, POVs, driver selections, assignments, and application preferences  
|                   |              | - Perform POV Copy, calculation, validation, data entry, and trace allocations  
|                   |              | - Deploy to Essbase and generate calculation scripts  
|                   |              | **Caution!** The Profitability and Cost Management Administrator must also be assigned Essbase access rights in order to perform Essbase ASO and BSO deployment.  
|                   |              | - Import and export data  
|                   |              | - Use the Lifecycle Management Utility to promote data from one environment, such as development or testing, to another environment, such as production.  
|                   |              | - Back up and restore Profitability and Cost Management model components.  
|                   |              | - Monitor changes made to business objects.  
|                   |              | - Create, edit, copy, delete, and launch queries.  
|                   |              | - Use Profitability Applications as an alternative to Performance Management Architect to create and manage new Profitability and Cost Management applications.  
| Power User        | Power        | - Create and maintain elements within the model, such as stages, drivers, POVs, driver selections, assignments, and application preferences.  
|                   |              | - Perform POV Copy, calculation, validation, data entry and trace allocations.  
|                   |              | - Import and export data  
|                   |              | - Deploy to Essbase and generate calculation scripts.  
|                   |              | **Caution!** The Profitability and Cost Management Power User must also be assigned Essbase access rights such as Create application and Application Manager in order to input data into Essbase and perform Essbase ASO and BSO deployment.  
|                   |              | - Create, edit, copy, delete, and launch queries.  
|                   |              | **Note:** The Power User does not necessarily require specific security roles to perform tasks. For example, if a Power User runs a calculation from the Calculate screen, this action creates and executes a taskflow behind the scenes. The Power User does not require the Manage Taskflow role to perform this task, unless the Power User wants to access this task directly from the Manage Taskflows task.  
| Interactive User  | Interactive  | - View all modelling screens  
|                   |              | - View and modify data in the Data Entry screen  
|                   |              | - View Trace Allocations  
|                   |              | - Launch queries.  


<table>
<thead>
<tr>
<th>Security Role</th>
<th>Type of Role</th>
<th>Description</th>
</tr>
</thead>
</table>
| View User     | Interactive  | View only access for these functions:  
  - Trace Allocations  
  - Application Preferences  
  - Model Stages, Drivers and POVs |
| Manage Taskflows | Shared Services Role | Required to create and edit taskflows.  
For more information, see the *Oracle Enterprise Performance Management System User Security Administration Guide*. |
| Run Taskflows | Shared Services Role | Required to enable users to only run and view taskflows. Users with this role cannot create or edit taskflows.  
For more information, see the *Oracle Enterprise Performance Management System User Security Administration Guide*. |
# Security Roles for Detailed Profitability

## Table 3  Detailed Profitability and Cost Management Security Roles

<table>
<thead>
<tr>
<th>Security Role</th>
<th>Type of Role</th>
<th>Description</th>
</tr>
</thead>
</table>
| Administrator       | Administrator| - Set up and maintain application preferences  
| (admin)             |              | - Build the model database using Performance Management Architect to select the common dimensions and members  
|                     |              | - Create and deploy reporting views to the relational database  
|                     |              | - Create, Read (View), Update and Delete the following functions:  
|                     |              |   - Stages  
|                     |              |   - Drivers  
|                     |              |   - POVs  
|                     |              |   - Driver Associations  
|                     |              |   - Assignments  
|                     |              |   - Application Preferences  
|                     |              |   - Calculation Rules  
|                     |              |   - Jobs Library and Status  
|                     |              |   - Table Registration  
|                     |              | - Perform the following tasks:  
|                     |              |   - POV Copy  
|                     |              |   - Validate  
|                     |              |   - Deploy  
|                     |              |   - Calculate  
|                     |              |   - Stop Jobs  
|                     |              | - Use the Lifecycle Management Utility to promote data from one environment, such as development or testing, to another environment, such as production.  
|                     |              | - Import and export data  
|                     |              | - Back up and restore Profitability and Cost Management model components.  
|                     |              | - Monitor changes made to business objects.  
|                     |              | - Deploy, update, and replace Essbase reporting databases, and transfer data.  
|                     |              | **Caution!** The Profitability and Cost Management Administrator must also be assigned Essbase access rights in order to perform Essbase ASO deployment.  
<p>|                     |              | - Use Profitability Applications as an alternative to Performance Management Architect to create and manage new Profitability and Cost Management applications. |</p>
<table>
<thead>
<tr>
<th>Security Role</th>
<th>Type of Role</th>
<th>Description</th>
</tr>
</thead>
</table>
| Power User     | Power        | - Create and maintain user accounts and security roles, and provision users, using Shared Services  
|                |              | - Create and deploy reporting views to the relational database  
|                |              | - Create, Read (View), Update and Delete the following functions:  
|                |              |   - Stages  
|                |              |   - Drivers  
|                |              |   - POVs  
|                |              |   - Driver Associations  
|                |              |   - Assignments  
|                |              |   - Application Preferences  
|                |              |   - Calculation Rules  
|                |              |   - Jobs Library and Status  
|                |              |   - Table Registration  
|                |              | - Perform the following tasks:  
|                |              |   - POV Copy  
|                |              |   - Validate  
|                |              |   - Calculate  
|                |              |   - Stop Jobs  
|                |              |   - Deploy, update, and replace Essbase Reporting databases, and transfer data.  
|                |              |   - Deploy  
|                |              | **Caution!** The Profitability and Cost Management Power User must also be assigned Essbase access rights such as Create application and Application Manager in order to input data into Essbase and perform Essbase ASO deployment.  
|                |              | **Note:** The Power User does not necessarily require specific security roles to perform tasks. For example, if a Power User runs a calculation from the Calculate screen, this action creates and executes a taskflow behind the scenes. The Power User does not require the manage Taskflow role to perform this task, unless the Power User wants to access this task directly from the Manage Taskflow task. |
| Interactive User | Interactive  | View (Read) the following functions:  
|                |              |   - Stages  
|                |              |   - Drivers  
|                |              |   - POVs  
|                |              |   - Driver Association  
|                |              |   - Assignments  
|                |              |   - Application Preferences  
|                |              |   - Calculation Rules  
|                |              |   - Jobs Library and Status  
|                |              |   - Table Registration  


<table>
<thead>
<tr>
<th>Security Role</th>
<th>Type of Role</th>
<th>Description</th>
</tr>
</thead>
</table>
| View User        | Interactive     | View (Read) the following functions:  
|                  |                 | - Stages  
|                  |                 | - Drivers  
|                  |                 | - POVs  
|                  |                 | - Driver Association  
|                  |                 | - Assignments  
|                  |                 | - Application Preferences  
|                  |                 | - Jobs Library and Status  
|                  |                 | - Table Registration  
| Manage Taskflows | Shared Services Role | Required to create and edit taskflows.  
|                  |                 | For more information, see the Oracle Enterprise Performance Management System User Security Administration Guide.  
| Run Taskflows    | Shared Services Role | Required to enable users to only run and view taskflows. Users with this role cannot create or edit taskflows.  
|                  |                 | For more information, see the Oracle Enterprise Performance Management System User Security Administration Guide.  

Security Roles for Management Ledger Profitability

Table 4  Standard Profitability and Cost Management Security Roles

<table>
<thead>
<tr>
<th>Security Role</th>
<th>Type of Role</th>
<th>Description</th>
</tr>
</thead>
</table>
| Administrator (admin)   | Power        | ● Create and maintain user accounts and security roles, and provision users, using Shared Services  
● Generate Essbase databases  
● Create and manage Profitability and Cost Management applications  
● Use Profitability Applications as an alternative to Performance Management Architect to create and manage new Profitability and Cost Management applications.  
● Set up and maintain application preferences  
● Build the model database using Performance Management Architect to select the common dimensions and members  
● Create and maintain elements within the model, such POVs, rule sets, and rules  
● Perform POV Copy, calculation, validation, data entry, and trace allocations  
● Deploy to Essbase and calculate models  
  **Caution!** The Profitability and Cost Management Administrator must also be assigned Essbase access rights in order to perform Essbase ASO deployment.  
● Import and export artifacts  
● Use the Lifecycle Management Utility to promote data from one environment, such as development or testing, to another environment, such as production  
● Back up and restore Profitability and Cost Management model components  
● Monitor changes made to business objects  
● Create, edit, copy, delete, and launch queries  
● Perform model balancing and validation |
| Power User              | Power        | ● Create and maintain elements within the model, such POVs, rule sets, and rules  
● Perform POV Copy, calculation, validation, data entry and trace allocations.  
● Import and export artifacts  
● Deploy to Essbase and calculate models  
● Create, edit, copy, delete, and launch queries  
● Perform model balancing and validation |
| Interactive User        | Interactive  | ● View all modeling screens  
● Use rule sets and rules  
● View Rule Balancing  
● View Trace Allocations  
● Define and run queries. |
| View User               | Interactive  | View only access for these functions:  
● Trace Allocations  
● Rule Balancing |
Auditing Changes in Profitability and Cost Management

You can monitor activity and changes in your application using the Audit feature in the 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 reports, as outlined in the following procedure. These reports can be exported as CSV files. See the Oracle Enterprise Performance Management System User Security Administration Guide.

➢ To enable auditing:

1. From EPM Workspace, select Navigate, then Administer, and then Shared Services Console.
2. From the Shared Services Console, 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 Under Select Tasks, 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.

**Note:** Most of these tasks apply only to Standard and Detailed Profitability applications.

**Table 5** Profitability and Cost Management Audit Tasks

<table>
<thead>
<tr>
<th>Monitored Area</th>
<th>Available Monitored Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stage</td>
<td>● Create stage&lt;br&gt;● Modify stage&lt;br&gt;● Delete stage&lt;br&gt;● Export stage&lt;br&gt;● Import stage</td>
</tr>
<tr>
<td>Driver</td>
<td>● Create driver&lt;br&gt;● Modify driver&lt;br&gt;● Delete driver&lt;br&gt;● Export driver&lt;br&gt;● Import driver</td>
</tr>
<tr>
<td>Monitored Area</td>
<td>Available Monitored Tasks</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>--------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| POV                           | • Create POV  
• Modify POV  
• Delete POV  
• Export POV  
• Import POV |
| Driver Selection              | • Create driver selection  
• Modify driver selection  
• Delete driver selection |
| Assignment                    | • Create assignment  
• Modify assignment  
• Delete assignment |
| Assignment Rules              | • Create assignment rule  
• Modify assignment rule  
• Delete assignment rule  
• Create rule selection  
• Remove rule selection |
| Cubes (Standard Profitability only) | • Deploy calculation cube  
• Deploy reporting cube  
• Transfer data |
| Genealogy (Standard Profitability only) | Calculate genealogy |
| Scripts                       | • Deploy allocation script  
• Deploy genealogy script  
• Deploy POV copy script  
• Execute allocation script  
• Execute genealogy script  
• Execute POV copy script |
| Group operations              | • Copy assignments  
• Delete assignments  
• Delete assignment rules' selections  
• Copy driver selections  
• Delete driver selection rules  
• Delete driver selection exceptions |
| Group import operations       | • Import stages  
• Import POVs  
• Import drivers  
• Import driver selections  
• Import assignments |
### Output Log Files

Administrators can generate log files throughout EPM Workspace to help technicians identify system or environmental problems, or to help developers debug reports or API programs.

The following log files are available for information concerning Profitability and Cost Management:

<table>
<thead>
<tr>
<th>Log File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hpcm.log</td>
<td>Profitability and Cost Management generates an application, server-side log file that collects application-specific messages that are sent from the application or server. By default, the log files are available at C:\oracle\Middleware\user_projects\domains\EPMS\servers\Profitability\logs. Contact your system administrator for access to this log file.</td>
</tr>
<tr>
<td>SharedServices_Security_Client.log</td>
<td>A Shared Services Client-side log file provides details regarding the Profitability and Cost Management handshake with Common Security Services. By default, the log file is available at C:\oracle\Middleware\user_projects\domains\EPMS\servers\Profitability\logs.</td>
</tr>
</tbody>
</table>

For additional log files for related products and applications, see the Oracle Enterprise Performance Management System Installation and Configuration Guide.

To modify the level of detail to be captured in the log files, see “Using System Logs” in the Oracle Hyperion Enterprise Performance Management System Installation and Configuration Troubleshooting Guide.
Oracle Diagnostic Logging (ODL) Files

The Oracle Diagnostic Logging (ODL) location for Web applications for all the configuration logging files for each EPM Workspace product are centralized.

For Profitability and Cost Management, the configuration logging file is available at:

C:\oracle\Middleware\user_projects\domains\EPMSystem\config\fmwconfig\servers\Profitability0. The file name is logging.xml.

For additional information on the configuration logging file, see “Using EPM Logs” in the Oracle Hyperion Enterprise Performance Management System Installation and Configuration Troubleshooting Guide.
In Working with Standard Profitability Applications:

- About Standard Profitability
- Working with Standard Profitability and Cost Management Dimensions and Members
- Managing Standard Profitability Applications and Dimensions Using Performance Management Architect
- Standard Profitability Import Staging Tables
- Exporting Model Definition Data for Standard Profitability
Understanding the Standard Profitability Product Architecture

Accessed through EPM Workspace, Profitability and Cost Management is an analytical tool 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.

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 Financial Reporting, Web Analysis, and Smart View.

### Working with Databases for Standard Profitability

For Profitability and Cost Management, you require both a relational database and an Essbase database. Details about these databases and connection information are specified on the System Information tab of the Model Summary in the application. For details, see the *Oracle Hyperion Profitability and Cost Management User’s Guide*.

When you deploy an application, Performance Management Architect populates tables with appropriate values in your Profitability and Cost Management relational database. The Essbase application is created during model deployment in Profitability and Cost Management. Each database stores specific information, as shown in Table 7, “Profitability and Cost Management Databases”.

<table>
<thead>
<tr>
<th>Database Type</th>
<th>Information Type Stored</th>
</tr>
</thead>
<tbody>
<tr>
<td>Essbase</td>
<td></td>
</tr>
<tr>
<td></td>
<td>● Dimension Hierarchies</td>
</tr>
<tr>
<td></td>
<td>● Cost Data</td>
</tr>
<tr>
<td></td>
<td>● Revenue Data</td>
</tr>
<tr>
<td></td>
<td>● Driver Data</td>
</tr>
<tr>
<td></td>
<td>● Currency Rates</td>
</tr>
<tr>
<td></td>
<td>● Calculation Scripts</td>
</tr>
<tr>
<td><strong>Database Type</strong></td>
<td><strong>Information Type Stored</strong></td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>------------------------------------------------------------------</td>
</tr>
<tr>
<td>Product Relational Database</td>
<td>● Dimensions</td>
</tr>
<tr>
<td></td>
<td>● Stage Definitions</td>
</tr>
<tr>
<td></td>
<td>● POV Definitions</td>
</tr>
<tr>
<td></td>
<td>● Driver Definitions</td>
</tr>
<tr>
<td></td>
<td>● Driver Selections</td>
</tr>
<tr>
<td></td>
<td>● Assignments</td>
</tr>
<tr>
<td></td>
<td>● Assignment Rules</td>
</tr>
<tr>
<td></td>
<td>● Assignment Rules Selections</td>
</tr>
<tr>
<td></td>
<td>● Model Preferences</td>
</tr>
</tbody>
</table>

Profitability and Cost Management uses the information from the driver definitions, driver selections and assignments defined in the relational database to create calculation scripts in the Essbase application. The calculation scripts perform the required data allocations.
About Standard Profitability Dimensions and Metadata

Dimensions and metadata are created and maintained by the Profitability and Cost Management Administrator, using Performance Management Architect or Profitability Applications. 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 Administration Guide or if using Profitability Applications, see Oracle Hyperion Profitability and Cost Management User’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:

- AllocationType dimension that stores both direct allocations and allocation genealogy data
- Dimensions that provide the structure for the model and scenarios, such as the Source and Destination Measures dimensions.
- Business dimensions that reflect the business-specific elements of the model, such as departments, General Ledger accounts, activities, customers, or products
- Point Of View (POV) dimensions, such as time periods, scenarios and versions
- Alias dimensions, as required
• 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 45.

• 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. For naming restrictions, see Appendix F, “Essbase Naming Conventions”.

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 Dimension Library. After deployment, the dimensions and members are available in the Profitability and Cost Management 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 “Setting the Dimension Sort Order” on page 56.

For detailed instructions on creating and maintaining the dimensions and members, see the Oracle Hyperion Enterprise Performance Management Architect Administration Guide. For naming conventions for dimensions and members, see the Appendix F, “Essbase Naming Conventions”.

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

• “Measures Dimensions” on page 46
• “AllocationType Dimension” on page 51
Dimension Types

- System dimensions, such as the Measures and AllocationType dimensions, must be created based on the requirements listed in these sections:
  - “Measures Dimensions” on page 46
  - “AllocationType Dimension” on page 51
- Alias dimension is used to assign alternate names, descriptions, languages, or other items that help to define dimensions.
- Business dimensions are created to describe the business elements within the model, such as departments, general ledger accounts, activities, customers, or products. See “Business Dimensions” on page 53.
- 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.
- User-defined attributes (UDAs) enable analysis based on text attributes that can be used to classify members of another associated dimension. For example, you may add a UDA called “New 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 maintain separate versions of the same POV. The version is used to track different versions of the same model and evaluate the impact of changes or different strategies on the model.

- Some Performance Management Architect dimension types are available for use in Profitability and Cost Management models:
  - Account
  - Entity
  - Country
  - Currency

See the Oracle Hyperion Enterprise Performance Management Architect Administration Guide for information on using these dimension types.

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.
**Measures Dimensions**

The Measures dimension contains the members required to build, validate and calculate a model. The Measures 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 Measures 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 Measures dimension type.

In addition to members required for the Measures dimension which **should not be modified**, users may add user-defined driver measures (or sub-hierarchies) to the hierarchy under the member 'UserDefinedDriverMeasures'. These dimension members can be modified by the user.

**Note:** You can give the Measures type dimension a different name, if required.

Within the Measures 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 46.
- Cost Layer Allocation Measures are used to control allocation of calculated and input costs. See “**Cost Layer Allocation Measures**” on page 47.
- Revenue Layer Allocation Measures are used to control the allocation of calculated and input revenue. See ”**Revenue Layer Allocation Measures**” on page 49.
- Reporting Measures are used to generate reports, using the calculated and input values to generate total costs and revenue for the model. All non-level-0 Reporting Measures are calculated. See “**Reporting Measures**” on page 50.

**Driver Measures**

**Table 8  Driver Measures**

<table>
<thead>
<tr>
<th>Member Name</th>
<th>Alias</th>
<th>Description</th>
<th>Calculated or Input</th>
</tr>
</thead>
<tbody>
<tr>
<td>FixedDriverValue</td>
<td>FixedDV</td>
<td>Default measure for use with driver types that require a fixed driver value parameter</td>
<td>Input</td>
</tr>
<tr>
<td>Rate</td>
<td>Rate</td>
<td>Default measure for use with driver types that require a rate parameter</td>
<td>Input</td>
</tr>
<tr>
<td>Quantity</td>
<td>Qty</td>
<td>Default measure for use with driver types that require a quantity parameter</td>
<td>Input</td>
</tr>
<tr>
<td>Weight</td>
<td>Weight</td>
<td>Default measure for use with driver types that require a weight parameter</td>
<td>Input</td>
</tr>
</tbody>
</table>
### Cost Layer Allocation Measures

**Table 9  Cost Layer Allocation Measures**

<table>
<thead>
<tr>
<th>Member Name</th>
<th>Alias</th>
<th>Description</th>
<th>Calculated or Input</th>
</tr>
</thead>
<tbody>
<tr>
<td>UnassignedCost</td>
<td>UnAsgCost</td>
<td>Remainder of cost at a source intersection after all assignment and idle calculations have been completed.</td>
<td>Calculated</td>
</tr>
<tr>
<td>CostAssigned</td>
<td>CostAsg</td>
<td>Total cost assigned from a source to post-stage destinations and non-reciprocal intrastage destinations</td>
<td>Calculated</td>
</tr>
<tr>
<td>CostAssignedIntraStage</td>
<td>CostAsgInt</td>
<td>Sum of costs assigned to intrastage destinations, excluding reciprocal destinations</td>
<td>Calculated</td>
</tr>
<tr>
<td>CostAssignedPostStage</td>
<td>CostAsgPost</td>
<td>Sum of costs assigned to post-stage destinations</td>
<td>Calculated</td>
</tr>
<tr>
<td><strong>Member Name</strong></td>
<td><strong>Alias</strong></td>
<td><strong>Description</strong></td>
<td><strong>Calculated or Input</strong></td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>OverDrivenCost</td>
<td>OverDrivenCost</td>
<td>For a standard basis driver, if the total cost assigned is greater than the NetCostForAssignment, then the overage amount is posted to <strong>OverDrivenCost</strong>.</td>
<td>Calculated</td>
</tr>
</tbody>
</table>
| IdleCost             | IdleCost        | Depending on the driver type, idle cost is generated differently:  
  - For actual basis drivers, idle cost is generated using the allocation formula: \( \text{IdleDriverValue}/\text{OverrideTotalDriverValue} \)  
  - For standard basis drivers, idle cost is generated if the total cost assigned is less than NetCostForAssignment.                                                                                                           | Calculated             |
| NetCostForAssignment | NetCostAsg      | Total cost available for assignment after accounting for all prior stage, intrastage, and reciprocal assignments  
  Set the DataStorage (BSO) property to **StoreData**.                                                                                                                                                                                                                                  | Calculated             |
| GrossReceivedCost    | GrRecCost       | Sum of all costs assigned from prior stages and intrastage assignments, excluding reciprocals and user input costs  
  Set the DataStorage (BSO) property to **StoreData**.                                                                                                                                                                                                                                  | Calculated             |
| StandardCostRate     | StandardCostRate| For a Standard Basis driver, the user assigns a standard cost rate, and enters that value for use in calculations for the standard basis cost driver, as follows: \( \text{CostReceivedPriorStage} = \text{StandardCostRate} \times \text{TotalDriverValue} \)                                                                                         | Input                  |
| 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  
  Set the DataStorage (BSO) property to **StoreData**.                                                                                                                                                                                                                               | 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 poststage and nonreciprocal intrastage destinations  
  Set the DataStorage (BSO) property to **StoreData**.                                                                                                                                                                                                                              | Calculated             |
<p>| 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             |</p>
<table>
<thead>
<tr>
<th>Member Name</th>
<th>Alias</th>
<th>Description</th>
<th>Calculated or Input</th>
</tr>
</thead>
<tbody>
<tr>
<td>CostPerDrvUnit</td>
<td>Cost Per Driver Unit</td>
<td>This measure is a child of AllocationMeasures. 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.</td>
<td>Calculated</td>
</tr>
<tr>
<td>UnitCost</td>
<td>Unit Cost</td>
<td>This measure is a child of AllocationMeasures. The formula uses the amount of cost at a source intersection (NetCostForAssignment) divided by a user-input quantity to calculate the cost per unit.</td>
<td>Calculated</td>
</tr>
</tbody>
</table>

### Revenue Layer Allocation Measures

#### Table 10  Revenue Layer Allocation Measures

<table>
<thead>
<tr>
<th>Member Name</th>
<th>Alias</th>
<th>Description</th>
<th>Calculated or Input</th>
</tr>
</thead>
<tbody>
<tr>
<td>UnassignedRevenue</td>
<td>UnAsgRev</td>
<td>Remainder of revenue at a source intersection after all assignment and idle calculations have been completed</td>
<td>Calculated</td>
</tr>
<tr>
<td>RevenueAssigned</td>
<td>RevAsg</td>
<td>Total Revenue assigned from a source to post-stage and non-reciprocal intrastage destinations</td>
<td>Calculated</td>
</tr>
<tr>
<td>OverDrivenRevenue</td>
<td>OverDrivenRevenue</td>
<td>For a standard basis driver, if the total revenue is greater than the NetRevenueForAssignment, then the overage amount is posted to OverDrivenRevenue.</td>
<td>Calculated</td>
</tr>
<tr>
<td>RevenueAssignedIntraStage</td>
<td>RevAsgInt</td>
<td>Sum of revenue assigned to intrastage destinations, excluding reciprocal destinations</td>
<td>Calculated</td>
</tr>
<tr>
<td>RevenueAssignedPostStage</td>
<td>RevAsgPos</td>
<td>Sum of revenue assigned to post-stage destinations</td>
<td>Calculated</td>
</tr>
<tr>
<td>IdleRevenue</td>
<td>IdleRev</td>
<td>Depending on the driver type, idle revenue is generated differently:</td>
<td>Calculated</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- For actual basis drivers, idle revenue is generated using the allocation formula: IdleDriverValue/OverrideTotalDriverValue.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- For standard basis drivers, revenue cost is generated if the total cost assigned is less than NetRevenueForAssignment.</td>
<td></td>
</tr>
<tr>
<td>NetRevenueForAssignment</td>
<td>NetRevAsg</td>
<td>Total revenue available for assignment after accounting for all prior stage, intrastage, and reciprocal assignments</td>
<td>Calculated</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Set the DataStorage (BSO) property to StoreData.</td>
<td></td>
</tr>
<tr>
<td>GrossReceivedRevenue</td>
<td>GrRecRev</td>
<td>Sum of all revenue assigned from prior stages and intrastage assignments, excluding reciprocal assignments and user input revenue</td>
<td>Calculated</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Set the DataStorage (BSO) property to StoreData.</td>
<td></td>
</tr>
</tbody>
</table>
### Member Name | Alias | Description | Calculated/Input
--- | --- | --- | ---
StandardRevenueRate | StandardRevenueRate | For a Standard Basis driver, the user assigns a standard revenue rate, and enters that value for use in calculations for the standard basis revenue driver, as follows: 
\[
\text{RevenueReceivedPriorStage} = \text{StandardRevenueRate} \times \text{TotalDriverValue}
\] | Input
RevenuInput | RevInput | User-entered revenue values for the intersection 
Define and store Revenue categories as a hierarchy under RevenueInput. 
Set the DataStorage (BSO) property to **StoreData**. | Input
RevenueReceived | RevRec | Sum of all revenue assigned to an intersection by prior stage and intrastage assignments, excluding revenue results from reciprocal assignments 
Set the DataStorage (BSO) property to **StoreData**. | Calculated
RevenueReceivedPriorStage | RevRecPri | Sum of revenue received on assignments from a prior stage | Calculated
RevenueReceivedIntraStage | RecRecInt | Sum of all revenue received on intrastage assignments, excluding reciprocal assignments | Calculated
NetReciprocalRevenue | NetRcpRev | Net effect of a reciprocal assignment on the amount of revenue available for assignment to poststage destinations and nonreciprocal intrastage destinations 
Set the DataStorage (BSO) property to **StoreData**. | Calculated
ReciprocalRevenueAssigned | RcpRevAsg | Total revenue assigned to reciprocal destinations, but excluding revenue received from the reciprocal assignment | Calculated
ReciprocalRevenueReceived | RcpRevRec | Total revenue received from a reciprocal destination | Calculated
ReciprocallIntermediateRevenue | RcpIntRev | Intermediate value calculated for an intersection after simultaneous equations have been applied, but before reciprocal adjustments have been made | Calculated

### Reporting Measures

**Table 11** Reporting Measures

<table>
<thead>
<tr>
<th>Member Name</th>
<th>Alias</th>
<th>Description</th>
<th>Calculated or Input</th>
</tr>
</thead>
</table>
| Profit | Profit | Value of the calculated profit for the selected intersection. This value is the result of the calculation: 
\[
\text{NetRevenueForAssignment} - \text{NetCostForAssignment}
\] | Calculated |
<table>
<thead>
<tr>
<th>Member Name</th>
<th>Alias</th>
<th>Description</th>
<th>Calculated or Input</th>
</tr>
</thead>
<tbody>
<tr>
<td>GrossCost</td>
<td>GrossCost</td>
<td>Total cost for an intersection, including all possible inputs:</td>
<td>Calculated</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Input values</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Prior stage assignments</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Intrastage assignments including reciprocals.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>This calculation represents the true total cost of the intersection.</td>
<td></td>
</tr>
<tr>
<td>StandardCost</td>
<td></td>
<td>For Standard Basis drivers, the calculated cost of StandardCostRate * TotalDriverValue</td>
<td>Calculated</td>
</tr>
<tr>
<td>StandardRevenue</td>
<td></td>
<td>For Standard Basis drivers, the calculated revenue of StandardRevenueRate * TotalDriverValue</td>
<td>Calculated</td>
</tr>
<tr>
<td>InitialCost</td>
<td>InitialCost</td>
<td>Cost of an intersection before intrastage or reciprocal costs are calculated,</td>
<td>Calculated</td>
</tr>
<tr>
<td></td>
<td></td>
<td>including input costs and cost received on assignments from prior stages.</td>
<td></td>
</tr>
<tr>
<td>NetCostAfterIntraStage</td>
<td>NetCostAftInt</td>
<td>Cost of an intersection, including all intrastage cost assignments</td>
<td>Calculated</td>
</tr>
<tr>
<td>GrossRevenue</td>
<td>GrossRev</td>
<td>Total revenue for an intersection, including all possible inputs:</td>
<td>Calculated</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Input values</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Prior stage assignments</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Intrastage assignments including reciprocals.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>This calculation represents the true total revenue for the intersection.</td>
<td></td>
</tr>
<tr>
<td>InitialRevenue</td>
<td>InitialRev</td>
<td>Revenue for an intersection before intrastage or reciprocal revenue is calculated, including input revenue, and revenue received on assignments from prior stages.</td>
<td>Calculated</td>
</tr>
<tr>
<td>NetRevenueAfterIntraStage</td>
<td>NetRevAftInt</td>
<td>Revenue for an intersection after accounting for all types of intrastage revenue assignments</td>
<td>Calculated</td>
</tr>
</tbody>
</table>

**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:** You can give the AllocationType type dimension a different name, 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.
This dimension 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 is available in the DirectAllocation member.

- **SysAllocVar2** is used to obtain a sum of the source links of DirectAllocation, Genealogy Allocation, and SysAllocVar1.

- **SysAllocVar3** stores calculated genealogy data that is used within the system. Do not use this member in reports.

- **TotalAllocation** dynamically calculates the sum of the source links of DirectAllocation, Genealogy Allocation, and SysAllocVar3.

- **IndirectAllocation** dynamically calculates the sum of the source links of Genealogy Allocation and SysAllocVar3.

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

### Alias Dimensions

Aliases are alternate names, descriptions, languages, or other items that help to define dimensions. For example, you may refer to a customer number in the system, but you can assign an alias that displays the company name on the screen, to make it easier to identify that client. You can assign one or more aliases to accounts, currencies, entities, scenarios, periods, versions, years, and user-defined dimension members.

**Note:** Duplicate member names or aliases are not allowed within the same dimension.

For Profitability and Cost Management, the alias must be set in Performance Management Architect. For detailed instructions on creating Alias dimensions, see the *Oracle Hyperion Enterprise Performance Management Architect Administration Guide*. 

52
Caution! If an Alias association is deleted in Performance Management Architect, it is not deleted from the model until the application is redeployed into Profitability and Cost Management.

When installation is complete, a “Default” alias table is available. After redeployment, you can view the alias on all screens that use the Common Member Selector, including Driver Selections and Assignments. To search and filter aliases, you must first select Show Alias from the Context Menu.

Note: The Alias View is not available on the Select Driver selector, which is accessed from Driver Definitions screen or from Driver Selections screen when adding or modifying a driver selection.

Aliases may be cloned if a particular dimension is cloned in Essbase.

To view Aliases:

1. From the application, select any screen that uses the Common Member Selector, for example, Driver Selections, Assignments, Data Entry, or Trace Allocations. The appropriate screen with Common Member Selector is displayed, showing all available members.

2. Optional: To select a member:
   - On Driver Selections, click Selector or Add .
   - On the Driver Rules and Exceptions tabs, click Add new exception, then select a dimension, then click Selector .
   - On Trace Allocations, click Selector .

   The Select Member dialog box opens, showing all available members.

3. In the Common Member Selector, click the Context Menu button and select Show Alias.

   Aliases are displayed in the list of members.

   Note: If you select Show Alias from the Context Menu, and no alias is assigned, the member name is displayed within square brackets. For example, the member name Product is displayed as [Product]

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 reuse 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 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 Dimension HierarchyType to “Enabled”, the first Gen2 member HierarchyType to “Stored” and the Gen2 member with alternative hierarchy with shared members to “Dynamic”

  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 need not be created for ADS or flat files, because it is added automatically by Performance Management Architect. Do not delete this member.

**POV Dimensions**

POV dimensions indicate a specific point of view or version of the model, such as year, scenario, or period. The dimension can be customized to reflect the requirements of your organization. For example, the POV may consist of quarters, months, seasonal groupings, and so on.
At least one POV dimension is required for each model, but you can create up to four POV dimensions. The POV dimensions are set in the Performance Management Architect Property grid for the model.

A Version dimension is also available, and is used to create another instance of your model. This version can be modified to enable you to experiment with strategies or business options to play “what-if” scenarios. By modifying the version, you can implement features, and compare results to determine the best course of action.

**Attribute Dimensions**

Attribute dimensions are a special type of dimension that are associated with a business dimension, and contain members that can be used to classify members of another, associated dimension. Attribute dimensions describe characteristics of data, such as the size and 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 assignment rules are created.

**Note:** For naming restrictions, see Appendix F, “Essbase Naming Conventions”.

Two Attribute dimension types exist in Performance Management Architect:

- **Attribute Dimensions:**
  - The attribute can be created using different structures, such as Boolean, Date, Numeric, and Text.
  - An attribute has a hierarchy, and the hierarchies can be rolled up for aggregate values.
  - Only one attribute from a given attribute dimension can be associated with one member.

- **User-Defined Attribute Dimensions (UDAs):**
  - The attribute can only be created using Text.
  - A UDA does not have a hierarchy and cannot be easily used in reports to generate sums.
  - Multiple UDAs can be associated with one member.

Each type of attribute dimension offers different advantages, depending on your model and reporting needs. For detailed information about working with attribute dimensions, see the *Oracle Essbase Database Administrator’s Guide*.

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.

For naming restrictions, see Appendix F, “Essbase Naming Conventions”.
**Dimension Sort Order and Density Settings**

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 a 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 56.

The Dimension Sort Order settings for the model are validated in Performance Management Architect. See “Modifying the Standard Profitability Application Settings” on page 73.

You can also set the density for dimensions from the Dimension Sort Order Settings dialog box.

### Dimension Sort Order Recommendations

Oracle recommends that you set the dimension sort using the following recommendations:

- 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, unique, and greater than or equal to 1.
- Measures 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 Recommendations” on page 56.

There are two ways to set the dimension sort order in Performance Management Architect:

- Using the Dimension Performance Settings dialog box to set the sort order for all dimensions at one time
Using the Property Grid in the Dimension Library to set individual dimension sort order for one dimension at a time

To set the Dimension Sort Order using the Dimension Performance Settings dialog box:

1. From EPM Workspace, select Navigate, then Administer, and then Dimension Library to display the Shared Library.
2. Right-click the name of the application, and select Dimension Performance Settings.
3. Select a dimension, and use the Up and Down arrows to move each dimension into the correct sort order, as described in “Setting the Dimension Sort Order” on page 56. The number of the original position for the dimension is displayed under Sort Order.
4. Optional: Under Density for the selected dimension, double-click the cell to display the options, and then select the appropriate density for the dimension. See “Optimizing Dimension Settings for Essbase” on page 60.
5. Click OK.

To set Dimension Sort Order for individual dimensions:

1. From EPM Workspace, select Navigate, then Administer, and then Dimension Library.
2. In the Profitability and Cost Management application, select the dimension for which you want to set the Dimension Sort Order.
3. In the Property Grid, select the Dimension Sort Order property, and under Value, enter the required number for the sort order. See “Setting the Dimension Sort Order” on page 56.
4. Click the Save icon.
Working with Standard Profitability Applications and Dimensions

Profitability and Cost Management uses Performance Management Architect or Profitability Applications 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.

**Note:** Lifecycle Management can be used not only to import or export model data, but also to import or export applications in Performance Management Architect.

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
Optimizing Dimension Settings for Essbase

A typical Standard 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 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. In that case, the AllocationType and Measures dimension should be set to Sparse.

Note: The block size should still remain within the limits of the Essbase best practices recommendations. See the Oracle Essbase Administration Services Developer’s Guide.

Standard Profitability 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. Set the associated Measures and AllocationType dimensions to Sparse. The decision 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.

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 Enterprise Performance Management System User Security Administration Guide.

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 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.

From the Dimension Library, you can edit the properties of these:

- 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 Administration 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 12, “Standard Profitability and Cost Management Dimension and Member Properties”, which displays the following information:

- The Property Label, which provides a more readable display name for the property. If applicable, the associated database type is appended to the name (ASO or BSO). If no database 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 Administration 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>
<thead>
<tr>
<th>Property Label</th>
<th>Description</th>
<th>Property Name</th>
</tr>
</thead>
</table>
| Alias         | Enter the alias for the selected dimension member. The alias is the alternate dimension member name that is displayed in a deployed application.  
  - To assign or change the default alias name for the current dimension member, click the existing alias name and enter the new one. Follow the proper naming rules. All databases have an alias table named Default.  
  - To assign or change an alias name for the current dimension member, click the existing alias name next to the alias table and enter the new name. Follow the proper naming rules. | |
| Application Type | Select the Application Type:  
  - General for Standard Profitability  
  - Detail for Detailed Profitability | ApplicationType |
| Attributes     | Enter the characteristics of a dimension member.  
  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 | Associate an attribute type with the selected member. Create attribute values for attributes assigned to dimension members that you can use to query on and to filter members.  
  Attribute types:  
  - Text  
  - Boolean  
  - Date  
  - Numeric  
  For example, the attribute type for “Caffeinated” would be Boolean; data values would be True or False. | 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.  
  You cannot assign a comment to an attribute dimension or member.  
  **Note:** For Unicode-enabled databases, a maximum of 80 characters is allowed. | Comment |
<table>
<thead>
<tr>
<th>Property Label</th>
<th>Description</th>
<th>Property Name</th>
</tr>
</thead>
</table>
| Consolidation | Member consolidation properties determine how children roll up into their parents.  
If the current member is not a dimension or an attribute, select the consolidation operator to assign to the member:  
- + (addition) - Default  
- - (subtraction)  
- * (multiplication)  
- / (division)  
- % (percent)  
- ~ (ignore during consolidation)  
- ^ (never consolidate)  
- NotUsed  
**Note:** Some restrictions exist regarding the use of consolidation operators in aggregate storage outlines. See the Oracle Essbase Database Administrator’s Guide. | Consolidation |
| Data Storage(ASO) | Select the ASO storage options for dimensions and members:  
- StoreData—Data is stored with the dimension.  
- ShareData—Data associated with this member can be shared. The ShareData property applies to the member only. The Dimension Root Member cannot be shared.  
- 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.  
**Note:** This option does not apply to stored hierarchies in aggregate storage outlines.  
- LabelOnly—No data is associated with this dimension. | ASODimensionDataStorage (for dimension root member)  
- ASOMemberDataStorage (for dimension members) |
| Data Storage(BSO) | Select the BSO storage options for dimensions and members:  
- StoreData—Data is stored with the dimension.  
- ShareData—Data associated with this member can be shared. The ShareData property applies to the member only. The Dimension Root Member cannot be shared.  
- 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.  
- LabelOnly—No data is associated with this dimension.  
- DynamicCalcAndStore—Data associated with this dimension is not calculated until requested by a user. After calculation, the data is stored.  
- 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. | BSODimensionDataStorage (for dimension root member)  
- BSOMemberDataStorage (for dimension members) |
| Dimension Formula(ASO) | Double-click the cell, and then click the Selector button to open the Memo Editor. Enter the formula, and then click OK.  
Use the appropriate ASO MDX syntax. | ASODimensionFormula (for dimension root member) |
<table>
<thead>
<tr>
<th>Property Label</th>
<th>Description</th>
<th>Property Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimension Formula (BSO)</td>
<td>Double-click the cell, and then click the Selector button to open the Memo Editor. Enter the formula, and then click OK. Use the appropriate BSO CALCULATOR syntax.</td>
<td>BSODimensionFormula (for dimension root member)</td>
</tr>
<tr>
<td>Dimension Solve Order</td>
<td>Enter the numeric value in the solution sequence for selected dimensions. For example, if this dimension is to be solved second, enter “2.”</td>
<td>DimensionSolveOrder</td>
</tr>
<tr>
<td>Dimension Sort Order</td>
<td>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. For example, if this dimension is to be the second in the Essbase outline, enter “2.” Dimension Sort Order must be set for every dimension in the model, except Alias and UDA dimensions. The dimension sort order must be sequential, unique, and greater than or equal to 1. Set the sort order as outlined in “Setting the Dimension Sort Order” on page 56.</td>
<td>DimensionSortOrder</td>
</tr>
</tbody>
</table>
| Dimension Storage Type | The Dimension Storage Type property applies only to Block Storage (BSO) databases in Essbase. Select the type of storage required for the dimension:  
- Dense  
- Sparse | DimensionStorageType |
| Hierarchy Type (Dimensions Only) | The Hierarchy Type property applies only to Aggregate Storage (ASO) databases in Essbase. The Reporting database is generated by Profitability and Cost Management. Set the type of hierarchy for the dimension:  
- Select STORED for any dimension members that use the following consolidation symbols:  
  - +ADDITION  
  - ~ IGNORE  
  **Note:** The no-consolidation or IGNORE (~) operator can only be used in a STORED hierarchy if the member's parent is set to LABEL_ONLY  
- Select DYNAMIC for dimension members that use any consolidation symbol, including ADDITION and IGNORE, or if the dimension member contains a formula.  
- Select ENABLED to support alternate hierarchies | DimensionHierarchyType |
| Hierarchy Type (Members Only) | Set the type of hierarchy for the member:  
- Stored (For first Gen2 child)  
- Dynamic (For Second Gen2 child onwards. The Second Gen2 member can host the alternate hierarchies)  
- None (Do Not Use) | HierarchyType |
<table>
<thead>
<tr>
<th>Property Label</th>
<th>Description</th>
<th>Property Name</th>
</tr>
</thead>
</table>
| Level Usage for Aggregation | This property applies to the Gen1 or Gen2 hierarchy members who host a stored hierarchy in an ASO outline.  
Set to DEFAULT  
For additional information on this property, see the Oracle Essbase Database Administrator's Guide. |                                        |
| Member Formula (ASO)      | Double-click the cell and then click the Selector button to open the Memo Editor. Enter the formula, and then click OK. Use the appropriate ASO MDX syntax.                                                                                                                                   | ASOMemberFormula                       |
| Member Formula (BSO)      | Double-click the cell and then click the Selector button to open the Memo Editor. Enter the formula, and then click OK. Use the appropriate BSO (BSO CALCULATOR) syntax.                                                                                                                     | BSOMemberFormula                       |
| Member Solve Order (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.”  
This property applies to ASO databases only.  
Members that have a solve order of 0 inherit the solve order of their dimension.  
Members with the same solve order are evaluated in the order in which their dimensions appear in the database outline, unless otherwise specified by the dimension sort order property.  
Members with no solve order are evaluated after members with a solve order. | MemberSolveOrder                       |
| 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.  
Select one of the following options:  
  - **Default** – The view selection engine is free to decide which levels to pick. This is the initial value for new hierarchies.  
  - **AllLevels** – The view selection engine considers all levels of the hierarchy for aggregation.  
  - **NoAggregation** – The view selection engine cannot select any levels for aggregation.  
  - **TopLevelOnly** – The view selection engine considers only the highest level of the dimension for aggregation.  
  - **BottomaTop** – The view selection engine considers only the highest level and the lowest level of the dimension for aggregation.  
  - **BottomLevelOnly** – The view selection engine considers only the bottom level of the dimension for aggregation. | PrimaryLevelWeighting                  |
<table>
<thead>
<tr>
<th>Property Label</th>
<th>Description</th>
<th>Property Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Two Pass Calculation</td>
<td>For BSO databases only, select the check box to calculate a member on the second pass through the outline.</td>
<td>TwoPassCalc</td>
</tr>
<tr>
<td>(Dimensions Only)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Two Pass Calculation</td>
<td>For BSO databases only, select the check box to calculate a member on the second pass through the outline.</td>
<td>TwoPassCalc</td>
</tr>
<tr>
<td>(Members Only)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UDA</td>
<td>Select the UDA member 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. You can select UDA member of up to 80 alphanumeric characters. <strong>Note:</strong> For Unicode-enabled databases, a maximum of 80 characters is allowed.</td>
<td>UDA</td>
</tr>
<tr>
<td>Unicode</td>
<td>Select <strong>Unicode</strong> to set the application to Unicode mode. Unicode-mode applications support multiple character sets.</td>
<td>Unicode</td>
</tr>
</tbody>
</table>

**Setting Hierarchy Type Property**

The ‘Hierarchy Type’ property applies only to Aggregate Storage Databases in Essbase. (The Reporting Cube is generated by Profitability and Cost Management).

To be able to use other consolidation type symbols (other than ADDITION and IGNORE) the hierarchy type should be set to ‘DYNAMIC.’

**Note:** If a dimension member has a formula, the hierarchy type must be set to ‘DYNAMIC.’

➢ To set the Hierarchy Type:

1. **From EPM Workspace,** select **Navigate,** then **Administer,** and then **Dimension Library.**
2. **Under the application,** select some dimension (except Alias, Attribute, UDA).
3. **In the Property Grid,** select the appropriate **Hierarchy Type** property:
   - Select **Stored** for any dimension whose dimension members use the following consolidation symbols:
     - + ADDITION
     - ~ IGNORE (only underneath LABEL ONLY members.)
   - Select **Dynamic** for any dimension whose dimension members needs to use all supported consolidation symbols, or have a formula.
4. **Redeploy the application to Profitability and Cost Management,** and then **Essbase.**
Creating Standard Profitability Applications

You can create Standard 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.

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.

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

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 special characters, including “&” (ampersands). See Appendix F, “Essbase Naming Conventions”

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

6. **Optional**: 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 Standard Profitability Dimensions Manually” on page 72.
8 **Optional:** To automatically create all required dimensions, select **Auto Create Local Dimensions**.

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 Standard Profitability Dimensions” on page 69.

### Selecting Standard Profitability Dimensions

After you create the application, you must select the dimensions that are to be included in the Standard Profitability application.

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

- Measures dimension contains the members required to build, validate and calculate a model, including driver measures, reporting measures, and allocation measures. Users may add 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.

  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 or color of products.

  There are two Attribute dimension types in Performance Management Architect:
Attribute dimensions:
- The attribute can be created using different structures, such as Boolean, Date, Numeric and Text.
- An attribute has a hierarchy, and the hierarchies can be rolled up for aggregate values.
- Only one attribute can be associated with one member.

User-defined attribute dimensions (UDAs):
- The attribute can only be created using Text.
- A UDA does not have a hierarchy, and cannot be easily used in reports to generate sums.
- Multiple UDAs can be associated with a member.

Each type of attribute dimension offers different advantages, depending on the requirements for your model and reporting needs. For detailed information about working with attribute dimensions, see Oracle Essbase Database Administrator's Guide.

- Alias is optional, and only required if you want to use aliases in your model. For naming conventions, see Appendix F, “Essbase Naming Conventions”.

Caution! Duplicate member names or aliases are not allowed within the same dimension.

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 Standard Profitability Dimensions Using the Wizard” on page 70.
- If you created a blank application, see “Adding Standard Profitability Dimensions Manually” on page 72.

Adding Standard Profitability 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:
- Measures Dimension
- AllocationType Dimension
- POV Dimension
• Alias Dimension (Optional)
• Business Dimensions
• Attribute Dimensions (Optional)

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).

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 view restricted words and characters for names, see Appendix F, "Essbase Naming Conventions".

➢ 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 required 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.

7 Click Next.

   The Application Settings screen is displayed.

8 Modify the application settings, as required. See “Modifying the Standard Profitability Application Settings” on page 73.

Adding Standard Profitability 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—Displays 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.
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 76.

6. **Modify the application settings, as required.** See “Modifying the Standard Profitability Application Settings” on page 73.

### Modifying the Standard Profitability Application Settings

**Subtopics**

- Modifying Application Settings Using the Dimension Library
- Modifying Application Settings Using the Application Wizard
- Setting the Properties for POV Dimensions
- Reordering Children Members

After you create and select the dimensions, you 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 74.
- Use the Application Wizard for auto-generated applications. See “Modifying Application Settings Using the Application Wizard” on page 74.

Modifying Application Settings Using the Dimension Library

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

- For a Profitability and Cost Management model, 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 75.
  - Set the Dimension Sort Order for all model dimensions.
    See “Setting the Dimension Sort Order” on page 56
  - Reorder the NoMember to display this member as the last generation 2 member on the list.
    See “Reordering Children Members” on page 76
- Modify dimensions, as required, using the procedures provided in the Oracle Hyperion Enterprise Performance Management Architect Administrator’s Guide:

Modifying Application 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.

Use the right-click menu to modify the application dimensions, and perform these tasks:

- For a Profitability and Cost Management model, 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 75.
  - Set the Dimension Sort Order for all model dimensions.
    See “Setting the Dimension Sort Order” on page 56.
- Reorder the NoMember to display this member as the last generation 2 member on the list.
  See “Reordering Children Members” on page 76
- Modify dimensions, as required, using the procedures provided in the Oracle Hyperion Enterprise Performance Management Architect Administration 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. In the Dimension Library, 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 Gen2 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 only required for business dimensions.

➢ To reorder children members:

1 From EPM Workspace, select Navigate, then Administer, and then Dimension Library to display the Shared Library.

2 Select the dimension to be reordered, as described below:

- For Shared dimensions, under Shared Library, right-click the dimension that you want to reorder and then select Reorder Children.
For Local dimensions, under the Application column, right-click the dimension that you want to reorder and then select **Reorder Children**.

The Reorder Children dialog box is displayed.

3. Under **Sort Order**, double-click the cell for which the member is to be reordered, and enter the number of the new sort order location.

   ![Reorder Children dialog box](image)

   **Note:** Ensure the NoMember dimension is moved to the end of the list as the last Gen2 member; otherwise, the deployment will fail.

4. Repeat step 3 to order all members.

5. To preview the changes, click **Preview**.

6. Click **Save**.

**Validating and Deploying the Standard Profitability Application in Performance Management Architect**

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

- “Profitability and Cost Management Validations” on page 78
- “Validating and Deploying Applications Using the Wizard” on page 79
- “Validating and Deploying Applications through Performance Management Architect” on page 80

**Note:** You can add, rename, reparent, or delete individual members, multiple members, or combinations of members during redeployment of an application.

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 Administration Guide*.

### Profitability and Cost Management Validations

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

#### Table 13  Profitability Validations

<table>
<thead>
<tr>
<th>Validation Level</th>
<th>Validations</th>
</tr>
</thead>
</table>
| Application      | - The name of the application must be seven characters or fewer, and cannot contain special characters.  
- At least one dimension must be set to POV type. Up to four dimensions may be marked as POV dimensions. Only one occurrence of each POV dimension class is allowed.  
- At least one business dimension must be defined.  
- At least one Measures dimension must be defined.  
- At least one AllocationType dimension must be defined.  
- There is only one dimension of type “Account.”  
- There is only one dimension of type “Entity.”  
- Application names do not include Essbase special characters and reserved words. |
<table>
<thead>
<tr>
<th>Validation Level</th>
<th>Validations</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dimension</strong></td>
<td></td>
</tr>
</tbody>
</table>
| • Root members of business dimensions must have the ASO and BSO data storage set to **LabelOnly**.  
| • Dimension Sort Order is set for all dimensions in the model, excluding Alias and UDA dimensions, and satisfies the following conditions:  
| | • A dimension sort order must be set for every dimension in the model, except Alias and UDA dimensions.  
| | **Note:** The Alias and UDA dimensions are ignored for Dimension Sort Order.  
| | • The dimension sort order must be sequential.  
| | • Measures dimension is set to 1 by default.  
| | • AllocationType dimension is set to 2 by default.  
| | • POV and business dimensions are set to 3 or higher.  
| | • Attribute dimensions are 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.  
| • Business dimensions should have at least two Gen 2 members.  
| • POV dimensions must have at least one member.  
| • Attribute dimensions must satisfy the following conditions:  
| | • Only a Level-0 member from the attribute dimension may be assigned as an attribute.  
| | • Attributes may be assigned only to members at the same level in the base dimension.  
| | • Attribute dimensions can only be associated with sparse dimensions. |
| **Member**        |  
| • Allow only ASO and BSO data storage to be defined.  
| • No shared members are allowed in the first Gen 2 member.  
| • A Shared Member must always appear after its corresponding Base Member in the outline order.  
| • **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 POVs, Measures, AllocationType, Alias, UDA, or Attribute dimensions.  
| • Shared members must reside in a Dynamic hierarchy, using one of the following methods:  
| | • The Gen 1 member is **HierarchyType=dynamic**  
| | • The Gen 1 member is **HierarchyType=HierarchiesEnabled**, and the Gen 2 ancestor of the shared member is **HierarchyType=dynamic**  
| • No duplicate member names or aliases exist for any members within the dimension.  
| • Member names do not include Essbase special characters and reserved words. |

**Validating and Deploying Applications Using the Wizard**

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.

➢ 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.

Validating and Deploying Applications through Performance Management Architect

To validate and 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 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 “Profitability and Cost Management Validations” on page 78.

When the validation is complete, the Job Task dialog box is displayed, showing the associated job number.

3 Click the link for the associated job to view the Job Console.

4 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.

5 Fix any errors listed in the Validation Log.

6 Repeat step 2 to step 5 until the validation succeeds with no errors.

The application is ready to be deployed.

7 In the Application Library, right-click the application name, and select Deploy from the drop-down list.

Applications are validated before they are deployed. Any validation errors display in the Job Console. You must correct all validation errors before deploying. When the validation is complete, the Deploy dialog box is displayed.
8 Provide the required information for the deployment:

- **Optional:** Under **Notes**, add notes pertaining to the deployment of this application. These notes are appended to the job description in the Job Console. This text field allows up to 200 characters and may contain any characters.

- 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 to which this application is to be assigned.

- **Optional:** Decide whether **Full Deploy** should be selected, as follows:
  - If this is the first deployment of the application, the **Full Deploy** checkbox is disabled because Performance Management Architect automatically sends all dimensions to Profitability and Cost Management.
  - If **Full Deploy** is checked for any subsequent deployment, then Performance Management Architect sends all dimensions to Profitability and Cost Management.
  - If **Full Deploy** is not checked, then Performance Management Architect sends only modified dimensions to Profitability and Cost Management. If no changes were made to a dimension, then that dimension is not included in the deployment.

9 **Click Deploy.**

The Job Task window is displayed, to confirm the Validation Job has been submitted, and display the Job ID.
10 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.

**Note:** If the deployment of the application is not successfully completed, from the Application Library, right-click the application name, and select Application Cleanup. This option forces a delete call to both Shared Services and Profitability and Cost Management to remove any traces of the application, and resets the application in Performance Management Architect to the undeployed state.

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

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

Data synchronization currently supports data transfer from BSO databases only.

**Note:** The Essbase application and database names should be the initial names given after the first Profitability and Cost Management application deployment. The database name should consist of the *application name* + “C” for the BSO database. If the names are different, data synchronization will fail.

Using Import 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 a database schema that is separate from the Profitability and Cost Management product schema. (You may re-use the same schema used as the Model Data Schema for Detailed Profitability). You then populate these staging tables with the details of the application artifacts to be imported.

Caution! Do not modify the product schema.

Staging database scripts are available for Microsoft SQL Server and Oracle Database. Use the appropriate script to create the import tables in a new database schema:

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

- “Creating Import Database Tables for Standard Profitability” on page 84
- “HPM_STG_STAGE” on page 84
- “HPM_STG_POV” on page 85
- “HPM_STG_DRIVER” on page 86
- “HPM_STG_DRIVER_SEL” on page 88
- “HPM_STG_DRIVER_EXCEP” on page 90
- “HPM_STG_ASSIGNMENT” on page 91
- “HPM_STG_ASGN_RULE_SEL” on page 93
Creating Import Database Tables for Standard Profitability

Staging database scripts are available for Microsoft SQL Server and Oracle Database. Use the create_staging.sql script to create the tables in Standard Profitability.

To create staging tables:

1. Create a new Oracle or Microsoft SQL Server database schema, outside of the product schema.
2. Navigate to the location for the appropriate SQL script in the default location for your database type:
   - %EPM_ORACLE_HOME%/products/Profitability/database/Common/MSSQLServer
   - %EPM_ORACLE_HOME%/products/Profitability/database/Common/Oracle
3. Run the create_staging.sql script.

**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 14 HPM_STG_STAGE**

<table>
<thead>
<tr>
<th>Field</th>
<th>SQL Server Data Type</th>
<th>Oracle Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>identity</td>
<td>integer(38,0)</td>
<td>Yes</td>
<td>Unique record ID</td>
</tr>
<tr>
<td>name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes</td>
<td>Name of the model stage.</td>
</tr>
<tr>
<td>description</td>
<td>nvarchar (255)</td>
<td>varchar2(255 CHAR)</td>
<td></td>
<td>Brief explanation of the purpose of the stage.</td>
</tr>
<tr>
<td>stage_order</td>
<td>identity (38,0)</td>
<td>integer(38,0)</td>
<td>Yes</td>
<td>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.</td>
</tr>
<tr>
<td>stage_prefix</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes</td>
<td>Differentiating prefix for the stage name.</td>
</tr>
<tr>
<td>dim1_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes</td>
<td>Name of the first dimension included in the stage.</td>
</tr>
<tr>
<td>dim2_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Name of the second dimension included in the stage, if available.</td>
</tr>
<tr>
<td>dim3_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Name of the third dimension included in the stage, if available.</td>
</tr>
<tr>
<td>Field</td>
<td>SQL Server Data Type</td>
<td>Oracle Data Type</td>
<td>Required</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>----------------------</td>
<td>------------------</td>
<td>----------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>driver_dim_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes</td>
<td>Name of the dimension selected as the driver for the stage.</td>
</tr>
<tr>
<td>Is_intra</td>
<td>nvarchar (1)</td>
<td>varchar2(255 CHAR)</td>
<td>Yes</td>
<td>Enter “Y” (Yes) to enable intrastage assignments within the model stage. Default is “N” (No). This field is populated from the application.</td>
</tr>
<tr>
<td>last_upload_date</td>
<td>timestamp (6)</td>
<td>timestamp (6)</td>
<td></td>
<td>The date and time of the last import.</td>
</tr>
<tr>
<td>exception</td>
<td>nvarchar (255)</td>
<td>varchar2(255 CHAR)</td>
<td></td>
<td>Message detailing errors that occurred during the import of this table.</td>
</tr>
<tr>
<td>created_userid</td>
<td>nvarchar (32)</td>
<td>varchar2(32 CHAR)</td>
<td></td>
<td>The ID of the user who initiated the last import.</td>
</tr>
<tr>
<td>created_timestamp</td>
<td>timestamp (6)</td>
<td>timestamp (6)</td>
<td></td>
<td>The date and time of the last import. This value is populated by the import program upon update.</td>
</tr>
<tr>
<td>modified_userid</td>
<td>nvarchar(32)</td>
<td>varchar2(32 CHAR)</td>
<td></td>
<td>The ID of the user who modified the last import.</td>
</tr>
<tr>
<td>modified_timestamp</td>
<td>timestamp (6)</td>
<td>timestamp (6)</td>
<td></td>
<td>The date and time of the modified import.</td>
</tr>
</tbody>
</table>

**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 15  HPM_STG_POV**

<table>
<thead>
<tr>
<th>Field</th>
<th>SQL Server Data Type</th>
<th>Oracle Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>identity</td>
<td>integer(38,0)</td>
<td>Yes</td>
<td>Unique record ID</td>
</tr>
<tr>
<td>pov_dim1_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes</td>
<td>Name of the first POV dimension member.</td>
</tr>
<tr>
<td>pov_dim2_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Name of the second POV dimension member, if available.</td>
</tr>
<tr>
<td>pov_dim3_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Name of the third POV dimension member, if available.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Additional dimensions and member names may be added, as required.</td>
</tr>
<tr>
<td>Field</td>
<td>SQL Server Data Type</td>
<td>Oracle Data Type</td>
<td>Required</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>----------------------</td>
<td>------------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>pov_dim4_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Name of the fourth POV dimension member, if available. Additional dimensions and member names may be added, as required.</td>
</tr>
<tr>
<td>pov_state</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes</td>
<td>The current status of the POV:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Draft</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Published</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Archived</td>
</tr>
<tr>
<td>last_upload_date</td>
<td>timestamp (6)</td>
<td>timestamp (6)</td>
<td></td>
<td>The date and time of the last import.</td>
</tr>
<tr>
<td>exception</td>
<td>nvarchar (255)</td>
<td>varchar2(255 CHAR)</td>
<td></td>
<td>Message detailing errors that occurred during the import of this table.</td>
</tr>
<tr>
<td>created_userid</td>
<td>nvarchar (32)</td>
<td>varchar2(32 CHAR)</td>
<td></td>
<td>The ID of the user who initiated the last import.</td>
</tr>
<tr>
<td>.created_timestamp</td>
<td>timestamp (6)</td>
<td>timestamp (6)</td>
<td></td>
<td>The date and time of the last import. This value is populated by the import program upon update.</td>
</tr>
<tr>
<td>modified_userid</td>
<td>nvarchar(32)</td>
<td>varchar2(32 CHAR)</td>
<td></td>
<td>The ID of the user who modified the last import.</td>
</tr>
<tr>
<td>modified_timestamp</td>
<td>timestamp (6)</td>
<td>timestamp (6)</td>
<td></td>
<td>The date and time of the modified import.</td>
</tr>
</tbody>
</table>

**HPM_STG_DRIVER**

The HPM_STG.Driver table provides details about the driver, including driver type, display order, layer and certain members from the Measures dimension.

Dependencies: None

**Table 16  HPM_STG_DRIVER**

<table>
<thead>
<tr>
<th>Field</th>
<th>SQL Server Data Type</th>
<th>Oracle Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>identity</td>
<td>integer(38,0)</td>
<td>Yes</td>
<td>Unique record ID</td>
</tr>
<tr>
<td>name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes</td>
<td>Name of the driver.</td>
</tr>
<tr>
<td>description</td>
<td>nvarchar (255)</td>
<td>varchar2(255 CHAR)</td>
<td>Yes</td>
<td>Description of the purpose for the selected driver.</td>
</tr>
<tr>
<td>Field</td>
<td>SQL Server Data Type</td>
<td>Oracle Data Type</td>
<td>Required</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>----------------------</td>
<td>------------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>driver_type</td>
<td>nvarchar (80)</td>
<td>varchar2 (80)</td>
<td>Yes</td>
<td>Type of driver:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Even</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Simple</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Simple Weighted</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Simple Variable</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Weighted Variable</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Fixed and Variable</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Percentage</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Custom</td>
</tr>
<tr>
<td>fixed_member</td>
<td>nvarchar (80)</td>
<td>varchar2 (80)</td>
<td></td>
<td>Member in the Measures dimension that corresponds to the standard driver</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>measure “FixedDriverValue.”</td>
</tr>
<tr>
<td>fixed_location</td>
<td>nvarchar (30)</td>
<td>varchar2 (30)</td>
<td></td>
<td>Location of the standard driver measure “FixedDriverValue.”</td>
</tr>
<tr>
<td>rate_member</td>
<td>nvarchar (80)</td>
<td>varchar2 (80)</td>
<td></td>
<td>Member in the Measures dimension that corresponds to the standard driver</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>measure “Rate.”</td>
</tr>
<tr>
<td>rate_location</td>
<td>nvarchar (30)</td>
<td>varchar2 (30)</td>
<td></td>
<td>Location of the standard driver measure “Rate.”</td>
</tr>
<tr>
<td>weight_member</td>
<td>nvarchar (80)</td>
<td>varchar2 (80)</td>
<td></td>
<td>Member in the Measures dimension that corresponds to the standard driver</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>measure “Weight.”</td>
</tr>
<tr>
<td>weight_location</td>
<td>nvarchar (30)</td>
<td>varchar2 (30)</td>
<td></td>
<td>Location of the standard driver measure “Weight.”</td>
</tr>
<tr>
<td>volume_member</td>
<td>nvarchar (80)</td>
<td>varchar2 (80)</td>
<td></td>
<td>Member in the Measures dimension that corresponds to the standard driver</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>measure “Volume.”</td>
</tr>
<tr>
<td>volume_location</td>
<td>nvarchar (30)</td>
<td>varchar2 (30)</td>
<td></td>
<td>Location of the standard driver measure “Volume.”</td>
</tr>
<tr>
<td>custom_formula</td>
<td>nvarchar (500)</td>
<td>varchar2 (500)</td>
<td></td>
<td>Formula created for the driver by a user. This formula must be created</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>using Essbase calculation script syntax. See the Oracle Hyperion Profitability</td>
</tr>
<tr>
<td>cost_layer</td>
<td>nvarchar (30)</td>
<td>varchar2 (30)</td>
<td></td>
<td>Enter “Yes” to set the driver layer to Cost. It contains the cost values</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>for the model.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> Either the cost_layer, revenue_layer, or both must be selected.</td>
</tr>
<tr>
<td>revenue_layer</td>
<td>nvarchar (30)</td>
<td>varchar2 (30)</td>
<td></td>
<td>Enter “Yes” to set the driver layer to Revenue. It contains the revenue</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>values for the model.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> Either the cost_layer, revenue_layer, or both must be selected.</td>
</tr>
<tr>
<td>Field</td>
<td>SQL Server Data Type</td>
<td>Oracle Data Type</td>
<td>Required</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>----------------------</td>
<td>------------------</td>
<td>----------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| allow_idle_cost    | nvarchar (1)         | varchar2(1 CHAR)|          | Enter the appropriate value to determine whether idle costs are allowed for this driver:  
  - Enter “Y” (Yes) to allow idle cost for a driver.  
  - Enter “N” (No) to disable idle cost for a driver. This is the default. |
| driver_basis_type  | nvarchar (80)        | varchar2(80 CHAR)| Yes      | Select the required driver basis:  
  - Actual Basis  
  - Standard Basis  
  **Note:** The Standard Basis driver cannot be used with Even or Percentage driver types. |
| priority           | identity             | integer(38,0)    | Yes      | Enter the calculation priority of a driver so that allocations within a stage can be run in the specified order.  
  The driver with the lowest priority is processed first. By default, the priority is set to 100, and the highest priority is 1. Numbers need not be sequential. Drivers with the same priority are executed in no particular order. Only whole, positive numbers are valid. |
| last_upload_date   | timestamp (6)        | timestamp (6)    |          | The date and time of the last import. |
| exception          | nvarchar (255)       | varchar2(255 CHAR)|          | Message detailing errors that occurred during the import of this table. |
| created_userid     | nvarchar (32)        | varchar2(32 CHAR)|          | The ID of the user who initiated the last import. |
| created_timestamp  | timestamp (6)        | timestamp (6)    |          | The date and time of the last import. This value is populated by the import program upon update. |
| modified_userid    | nvarchar(32)         | varchar2(32 CHAR)|          | The ID of the user who modified the last import. |
| modified_timestamp | timestamp (6)        | timestamp (6)    |          | The date and time of the modified import. |

**HPM_STG_DRIVER_SEL**

The HPM_STG_DRIVER_SEL table provides details about the POV and source stage driver dimension member for selected driver rules.
Dependencies:
- HPM_STG_DRIVER
- HPM_STG_STAGE
- HPM_STG_POV

Table 17   HPM_STG_DRIVER_SEL

<table>
<thead>
<tr>
<th>Field</th>
<th>SQL Server Data Type</th>
<th>Oracle Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>identity</td>
<td>integer(38,0)</td>
<td>Yes</td>
<td>Unique record ID</td>
</tr>
<tr>
<td>pov_dim1_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes</td>
<td>Name of the first dimension member in the selected point of view.</td>
</tr>
<tr>
<td>pov_dim2_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Name of the second dimension member in the point of view, if applicable.</td>
</tr>
<tr>
<td>pov_dim3_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Name of the third dimension member in the point of view, if applicable.</td>
</tr>
<tr>
<td>pov_dim4_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Name of the fourth dimension member in the point of view, if applicable.</td>
</tr>
<tr>
<td>driver_dim_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes</td>
<td>Name of the selected driver dimension member</td>
</tr>
<tr>
<td>layer_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes</td>
<td>Name of the selected layer for the point of view:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Cost Revenue</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Revenue</td>
</tr>
<tr>
<td>src_stage_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes</td>
<td>Name of the model stage for the selected point of view.</td>
</tr>
<tr>
<td>driver_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes</td>
<td>Name of the driver for the selected driver dimension member.</td>
</tr>
<tr>
<td>last_upload_date</td>
<td>timestamp (6)</td>
<td>timestamp (6)</td>
<td></td>
<td>The date and time of the last import.</td>
</tr>
<tr>
<td>import_exception</td>
<td>nvarchar (255)</td>
<td>varchar2 (255 CHAR)</td>
<td></td>
<td>Message detailing errors that occurred during the import of this table.</td>
</tr>
<tr>
<td>created_userid</td>
<td>nvarchar (32)</td>
<td>varchar2 (32 CHAR)</td>
<td></td>
<td>Populated by the import program upon update.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>The ID of the user who initiated the last import.</td>
</tr>
<tr>
<td>created_timestamp</td>
<td>timestamp (6)</td>
<td>timestamp (6)</td>
<td></td>
<td>Populated by the import program upon update.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>The date and time of the initiated the last import. This value is</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>populated by the import program upon update.</td>
</tr>
<tr>
<td>Field</td>
<td>SQL Server Data Type</td>
<td>Oracle Data Type</td>
<td>Required</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>----------------------</td>
<td>------------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>modified_userid</td>
<td>nvarchar(32)</td>
<td>varchar2 (32 CHAR)</td>
<td></td>
<td>Populated by the import program upon update.</td>
</tr>
<tr>
<td>modified_timestamp</td>
<td>timestamp (6)</td>
<td>timestamp (6)</td>
<td></td>
<td>Populated by the import program upon update.</td>
</tr>
</tbody>
</table>

**HPM_STG_DRIVER_EXCEP**

The HPM_STG_DRIVER_EXCEP table provides details about the POV and source stage intersection for the selected driver exceptions.

**Table 18  HPM_STG_DRIVER_EXCEP**

<table>
<thead>
<tr>
<th>Field</th>
<th>SQL Server Data Type</th>
<th>Oracle Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>identity</td>
<td>integer(38,0)</td>
<td>Yes</td>
<td>Unique record ID</td>
</tr>
<tr>
<td>pov_dim1_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes</td>
<td>Name of the first dimension member in the selected point of view.</td>
</tr>
<tr>
<td>pov_dim2_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Name of the second dimension member in the point of view, if applicable.</td>
</tr>
<tr>
<td>pov_dim3_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Name of the third dimension member in the point of view, if applicable.</td>
</tr>
<tr>
<td>pov_dim4_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Name of the fourth dimension member in the point of view, if applicable.</td>
</tr>
<tr>
<td>src_dim1_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes</td>
<td>Name of the first dimension member in the source stage</td>
</tr>
<tr>
<td>src_dim2_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Name of the second dimension member in the source stage, if applicable.</td>
</tr>
<tr>
<td>src_dim3_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Name of the third dimension member in the source stage, if applicable.</td>
</tr>
<tr>
<td>layer_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes</td>
<td>Name of the selected layer for the point of view:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>● Cost Revenue</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>● Revenue</td>
</tr>
<tr>
<td>src_stage_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes</td>
<td>Name of the source stage for the selected point of view.</td>
</tr>
</tbody>
</table>
### HPM_STG_ASSIGNMENT

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

**Dependencies:**
- HPM_STG_POV
- HPM_STG_STAGE
- HPM_STG_DRIVER

<table>
<thead>
<tr>
<th>Field</th>
<th>SQL Server Data Type</th>
<th>Oracle Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>identity</td>
<td>integer(38,0)</td>
<td>Yes</td>
<td>Unique record ID</td>
</tr>
<tr>
<td>pov_dim1_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes</td>
<td>Name of the first dimension member in the selected point of view.</td>
</tr>
<tr>
<td>pov_dim2_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Name of the second dimension member in the point of view, if applicable.</td>
</tr>
<tr>
<td>Field</td>
<td>SQL Server Data Type</td>
<td>Oracle Data Type</td>
<td>Required</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>----------------------</td>
<td>---------------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>pov_dim3_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Name of the third dimension member in the point of view, if applicable.</td>
</tr>
<tr>
<td>pov_dim4_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Name of the fourth dimension member in the point of view, if applicable.</td>
</tr>
<tr>
<td>layer_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes</td>
<td>Name of the selected layer for the point of view:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>*  Cost Revenue</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>*  Revenue</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> Either the Cost or the Revenue must be selected.</td>
</tr>
<tr>
<td>src_stage_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes</td>
<td>Name of the source stage for the assignment.</td>
</tr>
<tr>
<td>src_dim1_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes</td>
<td>Name of the first dimension member in the source stage.</td>
</tr>
<tr>
<td>src_dim2_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes, if stage source/destination stage have this dimension</td>
<td>Name of the second dimension member in the source stage, if available.</td>
</tr>
<tr>
<td>src_dim3_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes, if stage source/destination stage have this dimension</td>
<td>Name of the third dimension member in the source stage, if available.</td>
</tr>
<tr>
<td>dst_stage_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes</td>
<td>Name of the destination stage for the assignment.</td>
</tr>
<tr>
<td>dst_dim1_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes</td>
<td>Name of the first dimension member in the destination stage.</td>
</tr>
<tr>
<td>dst_dim2_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Name of the second dimension member in the destination stage, if available.</td>
</tr>
<tr>
<td>dst_dim3_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Name of the third dimension member in the destination stage, if available.</td>
</tr>
<tr>
<td>last_upload_date</td>
<td>timestamp (6)</td>
<td>timestamp (6)</td>
<td></td>
<td>The date and time of the last import.</td>
</tr>
<tr>
<td>import_exception</td>
<td>nvarchar (255)</td>
<td>varchar2 (255 CHAR)</td>
<td></td>
<td>Message detailing errors that occurred during the import of this table.</td>
</tr>
<tr>
<td>created_userid</td>
<td>nvarchar (32)</td>
<td>varchar2 (32 CHAR)</td>
<td></td>
<td>Populated by the import program upon update.</td>
</tr>
<tr>
<td>created_timestamp</td>
<td>timestamp (6)</td>
<td>timestamp (6)</td>
<td></td>
<td>The date and time of the initiated the last import.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>The value is populated by the import program upon update.</td>
</tr>
</tbody>
</table>
### Table 19  HPM_STG_ASIGN_RULE_SEL

<table>
<thead>
<tr>
<th>Field</th>
<th>SQL Server Data Type</th>
<th>Oracle Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>modified_userid</td>
<td>nvarchar(32)</td>
<td>varchar2 (32 CHAR)</td>
<td></td>
<td>The ID of the user who modified the last import.</td>
</tr>
<tr>
<td>modified_timestamp</td>
<td>timestamp (6)</td>
<td>timestamp (6)</td>
<td></td>
<td>The date and time of the modified import.</td>
</tr>
</tbody>
</table>

**HPM_STG_ASGN_RULE_SEL**

The HPM_STG_ASGN_RULE_SEL 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_ASGN_RULE_SEL**

<table>
<thead>
<tr>
<th>Field</th>
<th>SQL Server Data Type</th>
<th>Oracle Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>identity</td>
<td>integer(38,0)</td>
<td>Yes</td>
<td>Unique record ID</td>
</tr>
<tr>
<td>pov_dim1_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes</td>
<td>Name of the first dimension member in the selected point of view.</td>
</tr>
<tr>
<td>pov_dim2_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Name of the second dimension member in the point of view, if applicable.</td>
</tr>
<tr>
<td>pov_dim3_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Name of the third dimension member in the point of view, if applicable.</td>
</tr>
<tr>
<td>pov_dim4_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Name of the fourth dimension member in the point of view, if applicable.</td>
</tr>
<tr>
<td>layer_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes</td>
<td>Name of the selected layer for the point of view:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Cost Revenue</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Revenue</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> Either the Cost or the Revenue is required.</td>
</tr>
<tr>
<td>src_stage_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes</td>
<td>Name of the source stage for the assignment rule selection.</td>
</tr>
<tr>
<td>src_dim1_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes</td>
<td>Name of the first dimension member in the source stage.</td>
</tr>
<tr>
<td>Field</td>
<td>SQL Server Data Type</td>
<td>Oracle Data Type</td>
<td>Required</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>----------------------</td>
<td>----------------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>src_dim2_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes, if the source stage has the appropriate dimension.</td>
<td>Name of the second dimension member in the source stage, if available.</td>
</tr>
<tr>
<td>src_dim3_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes, if the source stage has the appropriate dimension.</td>
<td>Name of the third dimension member in the source stage, if available.</td>
</tr>
<tr>
<td>rule_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes</td>
<td>Name of the rule controlling the selected assignment. The rule must be present in the target database.</td>
</tr>
<tr>
<td>last_upload_date</td>
<td>timestamp (6)</td>
<td>timestamp (6)</td>
<td></td>
<td>Date of the last import.</td>
</tr>
<tr>
<td>import_exception</td>
<td>nvarchar (255)</td>
<td>varchar2 (255 CHAR)</td>
<td></td>
<td>Message detailing errors that occurred during the import of this table.</td>
</tr>
<tr>
<td>created_timestamp</td>
<td>timestamp (6)</td>
<td>timestamp (6)</td>
<td>Populated by the import program upon update.</td>
<td>The date and time of the last import.</td>
</tr>
<tr>
<td>created_userid</td>
<td>nvarchar (32)</td>
<td>varchar2 (32 CHAR)</td>
<td>Populated by the import program upon update.</td>
<td>ID of the user who initiated the last import.</td>
</tr>
<tr>
<td>modified_userid</td>
<td>nvarchar (32)</td>
<td>varchar2 (32 CHAR)</td>
<td>Populated by the import program upon update.</td>
<td>ID of the user who initiated the reimport.</td>
</tr>
<tr>
<td>modified_timestamp</td>
<td>timestamp (6)</td>
<td>timestamp (6)</td>
<td>Populated by the import program upon update.</td>
<td>The date and time of the reimport.</td>
</tr>
</tbody>
</table>
After the model has been created, you can query the database to display the model artifacts as output in a database view.

An Administrator can create 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_EXP_STAGE)
- POVs (HPM_EXP_POV)
- Drivers (HPM_EXP_DRIVER)
- Driver Selections (HPM_EXP_DRIVER_SEL)
- Driver Exceptions (HPM_EXP_DRIVER_EXCEP)
- Assignments (HPM_EXP_ASSIGNMENT)
- Assignment Rule Selections (HPM_EXP_ASGN_RULE_SEL)

### HPM_EXP_STAGE

The HPM_EXP_STAGE view retrieves the list of Stages defined within the model for all applications. This view also displays all the other attributes that are defined for the stage.

<table>
<thead>
<tr>
<th>Field</th>
<th>SQL Server Data Type</th>
<th>Oracle Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>application_name</td>
<td>nvarchar (80)</td>
<td>varchar2 (80)</td>
<td>Name of the application</td>
</tr>
<tr>
<td>id</td>
<td>identity</td>
<td>integer(38,0)</td>
<td>Stage ID saved in the Profitability and Cost Management database</td>
</tr>
<tr>
<td>Field</td>
<td>SQL Server Data Type</td>
<td>Oracle Data Type</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>----------------------</td>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the model stage.</td>
</tr>
<tr>
<td>description</td>
<td>nvarchar (255)</td>
<td>varchar2(255 CHAR)</td>
<td>Brief explanation of the purpose of the stage.</td>
</tr>
<tr>
<td>stage_order</td>
<td>integer</td>
<td>integer</td>
<td>The sequence position in which the selected stage is to be used within the</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>model while modeling (assignments and rules creation), in calc script</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>generation and calculation.</td>
</tr>
<tr>
<td>stage_prefix</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Differentiating prefix for the stage name.</td>
</tr>
<tr>
<td>dim1_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the first dimension included in the stage.</td>
</tr>
<tr>
<td>dim2_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the second dimension included in the stage, if available.</td>
</tr>
<tr>
<td>dim3_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the third dimension included in the stage, if available.</td>
</tr>
<tr>
<td>driver_dim_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the dimension selected as the driver for the stage.</td>
</tr>
<tr>
<td>ls_intra</td>
<td>nvarchar (1)</td>
<td>varchar2 (1 CHAR)</td>
<td>Flag determining whether intrastage assignments are allowed:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• “Y” (Yes) specifies that intrastage assignments within the model stage</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>are allowed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• “N” (No) specifies that intrastage assignments within the model stage</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>are not allowed.</td>
</tr>
</tbody>
</table>

**HPM_EXP_POV**

The HPM_EXP_POV view retrieves the states of each combination of dimension members included in a point of view (POV).

**Table 22  HPM_EXP_POV**

<table>
<thead>
<tr>
<th>Field</th>
<th>SQL Server Data Type</th>
<th>Oracle Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>application_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the application</td>
</tr>
<tr>
<td>id</td>
<td>identity</td>
<td>integer(38,0)</td>
<td>POV dimension member group ID saved in Profitability and Cost Management</td>
</tr>
<tr>
<td>pov_dim1_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the first POV dimension member.</td>
</tr>
<tr>
<td>pov_dim2_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the second POV dimension member, if available.</td>
</tr>
<tr>
<td>pov_dim3_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the third POV dimension member, if available. Additional</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>dimensions and member names may be added, as required.</td>
</tr>
<tr>
<td>Field</td>
<td>SQL Server Data Type</td>
<td>Oracle Data Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>----------------------</td>
<td>------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>pov_dim4_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the fourth POV dimension member, if available. Additional dimensions and member names may be added, as required.</td>
</tr>
<tr>
<td>pov_state</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>The current status of the POV: Draft, Published, Archived.</td>
</tr>
</tbody>
</table>

**HPM_EXP_DRIVER**

The HPM_EXP_DRIVER view retrieves the details about the driver, including driver type, formula and layer.

**Table 23  HPM_EXP_DRIVER**

<table>
<thead>
<tr>
<th>Field</th>
<th>SQL Server Data Type</th>
<th>Oracle Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>application_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the application</td>
</tr>
<tr>
<td>id</td>
<td>identity</td>
<td>integer</td>
<td>Driver ID saved in Profitability and Cost Management database</td>
</tr>
<tr>
<td>driver_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the driver.</td>
</tr>
<tr>
<td>description</td>
<td>nvarchar (255)</td>
<td>varchar2 (255 CHAR)</td>
<td>Description of the purpose for the selected driver.</td>
</tr>
<tr>
<td>display_order</td>
<td>integer</td>
<td>integer</td>
<td>Display position of the driver within the list of all drivers in the model.</td>
</tr>
<tr>
<td>driver_type</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Type of driver: Even, Simple, Simple Weighted, Simple Variable, Weighted Variable, Fixed and Variable, Percentage, Custom</td>
</tr>
<tr>
<td>complete_formula</td>
<td>nvarchar (500)</td>
<td>varchar2 (4000 CHAR)</td>
<td>Formula created for the driver by a user or a translated formula for pre-defined driver types. This formula must be created using Essbase calculation script syntax. See the Oracle Hyperion Profitability and Cost Management User's Guide.</td>
</tr>
<tr>
<td>Field</td>
<td>SQL Server Data Type</td>
<td>Oracle Data Type</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>----------------------</td>
<td>-----------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>cost_layer</td>
<td>nvarchar (30)</td>
<td>varchar2 (3 CHAR)</td>
<td>The driver layer is set to Cost layer, which contains the cost values for the model. Note: Either the cost_layer, revenue_layer, or both must be selected.</td>
</tr>
<tr>
<td>revenue_layer</td>
<td>nvarchar (30)</td>
<td>varchar2 (3 CHAR)</td>
<td>The driver layer is set to Revenue layer, which contains the revenue values for the model. Note: Either the cost_layer, revenue_layer, or both must be selected.</td>
</tr>
</tbody>
</table>
| allow_idle_cost | nvarchar(1)         | varchar2 (1 CHAR) | Flag determining whether idle costs are allowed for this driver:  
  * “Y” (Yes) specifying idle costs are allowed for this driver.  
  * “N” (No) specifying idle costs are not allowed for this driver. |
| driver_basis_type | nvarchar (80)     | varchar2(80 CHAR) | Select the required driver basis:  
  * Actual Basis  
  * Standard Basis  
  Note: The Standard Basis driver cannot be used with Even or Percentage driver types. |
| priority | integer              | integer(38,0)   | Enter the calculation priority of a driver so that allocations within a stage can be run in the specified order.  
  The driver with the lowest priority is processed first. By default, the priority is set to 100, and the lowest (or first) priority is 0. Numbers do not have to be sequential. Drivers with the same priority are executed in no particular order. Only whole, positive numbers are valid. |

**HPM_EXP_DRIVER_SEL**

The HPM_EXP_DRIVER_SEL view retrieves information about Driver Selection - Rules defined within the model for all applications.

Table 24  HPM_EXP_DRIVER_SEL

<table>
<thead>
<tr>
<th>Field</th>
<th>SQL Server Data Type</th>
<th>Oracle Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>application_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the application</td>
</tr>
<tr>
<td>id</td>
<td>identity</td>
<td>integer(38,0)</td>
<td>Unique record ID</td>
</tr>
<tr>
<td>pov_dim1_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the first dimension member in the selected point of view.</td>
</tr>
<tr>
<td>pov_dim2_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the second dimension member in the point of view, if applicable.</td>
</tr>
<tr>
<td>pov_dim3_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the third dimension member in the point of view, if applicable.</td>
</tr>
<tr>
<td>pov_dim4_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the fourth dimension member in the point of view, if applicable.</td>
</tr>
<tr>
<td>Field</td>
<td>SQL Server Data Type</td>
<td>Oracle Data Type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>----------------------</td>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>driver_dim_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the driver dimension member</td>
</tr>
<tr>
<td>layer_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the selected layer for the point of view:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>* Cost Revenue</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>* Revenue</td>
</tr>
<tr>
<td>src_stage_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the model stage for the selected point of view.</td>
</tr>
<tr>
<td>driver_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the driver for the selected driver dimension member.</td>
</tr>
</tbody>
</table>

**HPM_EXP_DRIVER_EXCEP**

The HPM_EXP_DRIVER_EXCEP view retrieves Driver Selection - Exceptions defined within the model for all applications.

**Table 25  HPM_EXP_DRIVER_EXCEP**

<table>
<thead>
<tr>
<th>Field</th>
<th>SQL Server Data Type</th>
<th>Oracle Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>application_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the application</td>
</tr>
<tr>
<td>id</td>
<td>identity</td>
<td>integer(38,0)</td>
<td>Driver selection Exception ID saved in the Profitability and Cost Management database.</td>
</tr>
<tr>
<td>pov_dim1_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the first dimension member in the selected point of view.</td>
</tr>
<tr>
<td>pov_dim2_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the second dimension member in the point of view, if applicable.</td>
</tr>
<tr>
<td>pov_dim3_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the third dimension member in the point of view, if applicable.</td>
</tr>
<tr>
<td>pov_dim4_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the fourth dimension member in the point of view, if applicable.</td>
</tr>
<tr>
<td>layer_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the selected layer for the point of view:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>* Cost Revenue</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>* Revenue</td>
</tr>
<tr>
<td>src_stage_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the model stage for the selected point of view.</td>
</tr>
<tr>
<td>src_dim1_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Member name of the first dimension of the selected stage.</td>
</tr>
<tr>
<td>src_dim2_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Member name of the second dimension of the selected stage, if applicable.</td>
</tr>
<tr>
<td>src_dim3_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Member name of the third dimension of the selected stage, if applicable.</td>
</tr>
</tbody>
</table>
Field   | SQL Server Data Type | Oracle Data Type | Description
---|---|---|---
driver_name | nvarchar (80) | varchar2(80 CHAR) | Name of the driver for the selected intersection.

**HPM_EXP_ASSIGNMENT**

The HPM_EXP_ASSIGNMENT view retrieves details about each assignment, including the source stage, POV, layer, source dimension members, destination stage, and destination dimension members.

**Table 26  HPM_EXP_ASSIGNMENT**

<table>
<thead>
<tr>
<th>Field</th>
<th>SQL Server Data Type</th>
<th>Oracle Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>application_name</td>
<td>nvarchar (80)</td>
<td>varchar2 (80)</td>
<td>Name of the application</td>
</tr>
<tr>
<td>id</td>
<td>identity</td>
<td>integer(38,0)</td>
<td>Assignment ID saved in Profitability and Cost Management database</td>
</tr>
<tr>
<td>pov_dim1_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2 (80)</td>
<td>Name of the first dimension member in the selected point of view</td>
</tr>
<tr>
<td>pov_dim2_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2 (80)</td>
<td>Name of the second dimension member in the point of view, if applicable</td>
</tr>
<tr>
<td>pov_dim3_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2 (80)</td>
<td>Name of the third dimension member in the point of view, if applicable</td>
</tr>
<tr>
<td>pov_dim4_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2 (80)</td>
<td>Name of the fourth dimension member in the point of view, if applicable</td>
</tr>
</tbody>
</table>
| layer_name                       | nvarchar (80)        | varchar2 (80)   | Name of the selected layer for the point of view:
  - Cost
  - Revenue |
| src_stage_name                   | nvarchar (80)        | varchar2 (80)   | Name of the source stage for the assignment |
| src_dim1_member_name             | nvarchar (80)        | varchar2 (80)   | Name of the first dimension member in the source stage |
| src_dim2_member_name             | nvarchar (80)        | varchar2 (80)   | Name of the second dimension member in the source stage, if available |
| src_dim3_member_name             | nvarchar (80)        | varchar2 (80)   | Name of the third dimension member in the source stage, if available |
| dst_stage_name                   | nvarchar (80)        | varchar2 (80)   | Name of the destination stage for the assignment |
| dst_dim1_member_name             | nvarchar (80)        | varchar2 (80)   | 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 |
### HPM_EXP_ASGN_RULE_SEL

The HPM_EXP_ASGN_RULE_SEL view retrieves details about the source stage and dimension members for a rule controlling the assignments for the selected stage.

<table>
<thead>
<tr>
<th>Field</th>
<th>SQL Server Data Type</th>
<th>Oracle Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dst_dim3_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2 (80)</td>
<td>Name of the third dimension member in the destination stage, if available.</td>
</tr>
<tr>
<td>application_name</td>
<td>nvarchar (80)</td>
<td>varchar2 (80)</td>
<td>Name of the application</td>
</tr>
<tr>
<td>id</td>
<td>identity</td>
<td>integer(38,0)</td>
<td>Unique record ID</td>
</tr>
<tr>
<td>pov_dim1_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2 (80)</td>
<td>Name of the first dimension member in the selected point of view.</td>
</tr>
<tr>
<td>pov_dim2_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2 (80)</td>
<td>Name of the second dimension member in the point of view, if applicable.</td>
</tr>
<tr>
<td>pov_dim3_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2 (80)</td>
<td>Name of the third dimension member in the point of view, if applicable.</td>
</tr>
<tr>
<td>pov_dim4_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2 (80)</td>
<td>Name of the fourth dimension member in the point of view, if applicable.</td>
</tr>
<tr>
<td>layer_name</td>
<td>nvarchar (80)</td>
<td>varchar2 (80)</td>
<td>Name of the selected layer for the point of view:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● Cost</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>● Revenue</td>
</tr>
<tr>
<td>src_stage_name</td>
<td>nvarchar (80)</td>
<td>varchar2 (80)</td>
<td>Name of the source stage for the assignment rule.</td>
</tr>
<tr>
<td>src_dim1_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2 (80)</td>
<td>Name of the first dimension member in the source stage.</td>
</tr>
<tr>
<td>src_dim2_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2 (80)</td>
<td>Name of the second dimension member in the source stage, if available.</td>
</tr>
<tr>
<td>src_dim3_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2 (80)</td>
<td>Name of the third dimension member in the source stage, if available.</td>
</tr>
<tr>
<td>rule_name</td>
<td>nvarchar (80)</td>
<td>varchar2 (80)</td>
<td>Name of the assignment rule controlling the selected assignment source combination. The Assignment Rule should already be defined in the Profitability and Cost Management database.</td>
</tr>
</tbody>
</table>
P a r t  I I I

Working with Detailed Profitability Applications

In Working with Detailed Profitability Applications:

- Understanding the Detailed Profitability Product Architecture
- Working with Detailed Profitability Databases
- Product Schema
- Model Data Schema for Detailed Profitability
- Working with Detailed Profitability and Cost Management Dimensions and Metadata
- Managing Detailed Profitability Applications and Dimensions using Performance Management Architect
- Calculating Detailed Profitability Models
- Importing Detailed Profitability Staging Tables
- Exporting Model Definition Data for Detailed Profitability
Accessed through EPM Workspace, Detailed Profitability is an analytical application that uses data in a relational database. The application enables business users to model their business for profitability and cost management, and use that model information to create a relational database where model information, existing data, and results calculated by the application are stored. Data, both entered and calculated, for the Detailed Profitability model is housed in relational databases.

Profitability and Cost Management leverages Performance Management Architect and Shared Services for the centralized management of application metadata and security (Figure 2 on page 105).

Figure 2  Detailed Profitability Architecture

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 SQL Statements required to perform calculations within the database. Each model requires access to the following schemas within the database:
- A schema, referred to as the Product Schema, for storing the model design including the dimension metadata deployed from Performance Management Architect. See Chapter 10, “Product Schema”.

- A schema, referred to as the Model Data Schema, for storing existing data and the results of the model calculations. See Chapter 11, “Model Data Schema for Detailed Profitability”.

For Detailed Profitability applications, your dimensional data and model definition are stored in the same relational database schema that is used to store dimensional data and model definitions for Standard Profitability applications. This schema is referred to as the Product Schema, and it is created when Profitability and Cost Management is installed. Dimensional data is populated in the Product Schema when you deploy your application from Performance Management Architect. Model definitions are stored in this schema as you build your model.

For Detailed Profitability applications, the business data upon which allocations are performed is also stored in the relational database (not in Essbase as is the case for Standard Profitability applications). This data resides in a separate database schema called the Model Data Schema. The Model Data Schema is user-defined and must reside in the same database instance as the Product Schema. Only Oracle and Microsoft SQL Server databases are supported.

After deploying the application from Performance Management Architect, the dimensions and members are mapped to columns and registered to the application tables in Profitability and Cost Management. For information on Detailed Profitability dimensions, see Chapter 12, “Working with Detailed Profitability and Cost Management Dimensions and Metadata”.

<table>
<thead>
<tr>
<th>Schema Type</th>
<th>Information Type Stored</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model Data Schema</td>
<td>• Dimension Hierarchies</td>
</tr>
<tr>
<td></td>
<td>• Cost Data</td>
</tr>
<tr>
<td></td>
<td>• Revenue Data</td>
</tr>
<tr>
<td></td>
<td>• Driver Data</td>
</tr>
<tr>
<td></td>
<td>• Profit Object Data</td>
</tr>
<tr>
<td></td>
<td>• Currency Rates</td>
</tr>
<tr>
<td>Product Schema</td>
<td>• Dimensions</td>
</tr>
<tr>
<td></td>
<td>• Stage Definitions</td>
</tr>
<tr>
<td></td>
<td>• POV Definitions</td>
</tr>
<tr>
<td></td>
<td>• Driver Definitions</td>
</tr>
<tr>
<td></td>
<td>• Driver Selections</td>
</tr>
<tr>
<td></td>
<td>• Assignment Rules</td>
</tr>
<tr>
<td></td>
<td>• Assignment Rules Selections</td>
</tr>
<tr>
<td></td>
<td>• Model Preferences</td>
</tr>
<tr>
<td></td>
<td>• Table Registrations</td>
</tr>
<tr>
<td></td>
<td>• Calculation Rules</td>
</tr>
</tbody>
</table>
Standard and Detailed Profitability applications share the same Product Schema that holds the Profitability and Cost Management system tables and that is created during installation. The Profitability and Cost Management UI and the Detailed Profitability calculation engine connect to this schema. Reporting views are created in the Product schema during model development and model calculation.

Access to this schema is typically restricted to application administrators and Database Administrators. Grants on the reporting views are automatically made to the Model Data Schema and synonyms for them are automatically created in the Model Data Schema. The synonyms expose them to users who have access to that schema without having to grant them privileges on the Product Schema.

After installation, Detailed Profitability requires that these system privileges (or roles that include them) be granted to the Product Schema:

- For Oracle database, Detailed Profitability requires that these system privileges (or roles that include them) be granted to the Product Schema:
  - CREATE TABLE
  - CREATE VIEW
  - CREATE ANY SYNONYM
  - DROP ANY SYNONYM

- For MSSQL Server, Detailed Profitability requires that these system privileges (or roles that include them) be granted to the Product Schema:
  - CREATE TABLE
  - CREATE VIEW
  - ALTER or CONTROL on the Model Data Schema to the product schema (so that the product schema has the ability to change objects in the Model Data schema)

**EXCEPTION:** If the Product Schema is an Administrator user such as “dbo” or “sa” it will already have the above privileges. In this case it is not necessary to perform the following grants.

**Note:** Do NOT grant the SYSADMIN role to the user associated with the HPCM product schema. It will change the default schema for that user, which must remain set as the HPCM product schema.
The CREATE TABLE and CREATE VIEW privileges allow Profitability and Cost Management to create system-generated tables and reporting views in the Product Schema during model development and during model calculation.

The CREATE ANY SYNONYM and DROP ANY SYNONYM privileges allow Profitability and Cost Management to manage the synonyms in the Model Data Schema for the system-generated reporting views that are created in the Product Schema.

Note: If granting either of these system privileges is a security concern for your organization, you may withhold these privileges and manually create the synonyms in the Model Data Schema for the system-generated reporting views. This option enables you to expose them to business data users who only have access to the Model Data Schema.
The Model Data Schema can be a pre-existing schema or one created by your Database Administrator to support your Detailed Profitability applications. This schema holds the tables and views that contain the business data on which your Detailed Profitability application performs allocations. Profitability and Cost Management only connects directly to the Product Schema; to see the tables or views in your application, the Database Administrator must grant table privileges on each one to the Product Schema.

Model Data Schema Table Grants to the Product Schema

In order to view database objects from the Model Data Schema within Model Data Registration, you must first grant table privileges to the Profitability and Cost Management Product Schema. Grant the privileges for each table type as shown on the table below:

<table>
<thead>
<tr>
<th>Table Type</th>
<th>Privileges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Stage Table</td>
<td>GRANT SELECT ON &lt;table_name or view_name&gt; TO &lt;hpm_product_schema&gt; WITH GRANT OPTION</td>
</tr>
<tr>
<td>Destination Stage Table</td>
<td>GRANT SELECT, UPDATE ON &lt;table_name or view_name&gt; TO &lt;hpm_product_schema&gt;</td>
</tr>
<tr>
<td>All types of Lookup Tables</td>
<td>GRANT SELECT ON &lt;table_name or view_name&gt; TO &lt;hpm_product_schema&gt;</td>
</tr>
</tbody>
</table>

Note: For Oracle database, you must grant EXECUTE privilege on dbms_stats. By default, this privilege is granted to Public. If your organization has restricted the collection of statistics and has revoked this privilege from public, then you must grant it to the product schema.
After deploying the application from Performance Management Architect, you must register the tables in the Model Data Schema for use in your Detailed Profitability application.

You map their columns to the dimensions and measures that you defined in Performance Management Architect. For information on Detailed Profitability dimensions, see Chapter 12, "Working with Detailed Profitability and Cost Management Dimensions and Metadata".

Model Data Schema Table Structural Requirements

To register a table from the Model Data Schema for use in a Detailed Profitability application, you must meet the following requirements:

- **Database object names**: Must be in upper case characters, and may include digits 0-9, and characters “_” and “$”. Lower case and other special characters are not supported. If you have existing database object names using lower case or special characters, you can create views to rename table, view, and column names using allowable identifiers and then register the views instead of their underlying tables.

- **Destination Stage tables only**: Must have a numeric column named “WORKING” (in upper case characters).

- **Destination Stage tables only**: Must have a primary key constraint defined on it to enforce the uniqueness of the table.

Registering Database Views

If you made the appropriate grants to the Product Schema, you may also register database views from the Model Data Schema. Any valid database view can be registered as the Source stage table or as a Lookup table. Only simple single-table updatable views may be registered as the Destination stage table.

Database views can be useful for several reasons:

- Accessing tables that do not reside in the Model Data Schema. This may especially be useful for lookup tables.

- Filtering rows or hiding columns in the underlying table to prevent exposure to the Detailed Profitability application.

- Hiding complex joins within the view definition for read-only tables to simplify their use within the application.

During model development and model calculation, Profitability and Cost Management creates system-generated reporting views in the Product Schema. Profitability and Cost Management also grants SELECT privilege on them to the Model Data Schema and attempts to create synonyms for them in the Model Data Schema. This provides users with access to that schema without having privileges on the Product Schema that are usually reserved for administrator access only. Model builders and report developers should normally be given access to this schema unless the administrator is expected to complete all the table registration tasks that Profitability and Cost Management requires, and the physical model setup in any reporting tools used.
Note: In Oracle, a database user and a database schema are the same thing; however, in Microsoft SQL Server, they are two separate items. If you are using Microsoft SQL Server, you must create a user with the same name as the Model Data Schema, and grant the Model Data Schema access to that user. The grants on system-generated views made by Profitability and Cost Management reference the Model Data User. The synonyms that are created go into the Model Data Schema.

For SQL Server, references to the Model Data Schema in the Profitability and Cost Management product and documentation refer to either the schema, or the user, or both.

Note: SQL Server: You must create the measure columns on your destination stage table and lookup tables with 4 or more decimal places in order to avoid rounding errors.

Caution! Before you import data or artifacts, you must create a backup of your Model Data Schema. See the Oracle Enterprise Performance Management System Backup and Recovery Guide
In This Chapter

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>About Detailed Profitability Dimensions and Metadata</td>
<td>115</td>
</tr>
<tr>
<td>Detailed Profitability Dimension Types</td>
<td>117</td>
</tr>
<tr>
<td>Alias Dimensions</td>
<td>117</td>
</tr>
<tr>
<td>Business Dimensions</td>
<td>118</td>
</tr>
<tr>
<td>POV Dimensions</td>
<td>119</td>
</tr>
<tr>
<td>Attribute Dimensions</td>
<td>120</td>
</tr>
<tr>
<td>Dimension Sort Order and Density Settings</td>
<td>120</td>
</tr>
</tbody>
</table>

About Detailed Profitability Dimensions and Metadata

Detailed Profitability enables you to use your existing relational database as the starting point for your Detailed Profitability application.

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 Administration 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. During creation of the Detailed Profitability application, the following dimensions are selected.

- MeasuresDetailed (Required)
- At least one Performance Management Architect POV dimension (Required)
- At least one Business Dimension (Required)
- Alias Dimension (Optional)
- Attribute Dimensions (Optional)

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 45.

- 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. For naming restrictions, see Appendix F, “Essbase Naming Conventions”.

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.

- MeasuresDetailed is a reserved dimension that contains the Allocation members require to support stage balancing and other verification activities. This dimension does not contain Driver measures.

- Dimensions that provide the structure for the model and scenarios, such as the Measures dimensions.

- Business dimensions that reflect the business-specific elements of the model, such as departments, General Ledger accounts, activities, customers, or products

- Point Of View (POV) dimensions, such as time periods, scenarios and versions

- Alias dimensions, as required

- Attribute dimensions, as required

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 Detailed Profitability and Cost Management model, they are automatically available in the Dimension Library. After deployment, the dimensions and members are available in the Profitability and Cost Management 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. Creating additional dimensions or members after the application has already been deployed may require re-registration in Model Data Registration in order for the new objects to be used.
For detailed instructions on creating and maintaining the dimensions and members, see the *Oracle Hyperion Enterprise Performance Management Architect Administration Guide*.

**Detailed Profitability Dimension Types**

For Detailed Profitability and Cost Management, these types of dimensions are available:

- **MeasuresDetailed** is a restricted dimension that contains the Allocation members required to support stage balancing and other verification activities.

- **Alias dimension** is used to assign alternate names, descriptions, languages, or other items that help to define dimensions. This is an optional dimension, and only required if you want to use aliases in your model.

  **Note:** Duplicate member names or aliases are not allowed.

- **Business dimensions** are created to describe the business elements within the model, such as departments, general ledger accounts, activities, customers, or products. At least one business dimension is required for the application.

- **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.

- **Some Performance Management Architect dimension types** are available for use in Profitability and Cost Management models:
  - Account
  - Entity
  - Country
  - Currency

  See the *Oracle Hyperion Enterprise Performance Management Architect Administration Guide* for information on using these dimension types.

  **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*.

**Alias Dimensions**

Aliases are alternate names, descriptions, languages, or other items that help to define dimensions. For example, you may refer to a customer number in the system, but you can assign an alias that displays the company name on the screen, to make it easier to identify that client. You can assign one or more aliases to accounts, currencies, entities, scenarios, periods, versions, years, and user-defined dimension members.

**Note:** Duplicate member names or aliases are not allowed within the same dimension.
For Profitability and Cost Management, the alias must be set in Performance Management Architect. For detailed instructions on creating Alias dimensions, see the Oracle Hyperion Enterprise Performance Management Architect Administration Guide.

**Caution!** If an Alias association is deleted in Performance Management Architect, it is not deleted from the model until the application is redeployed into Profitability and Cost Management.

When installation is complete, a “Default” alias table is available. After redeployment, you can view the alias on all screens that use the Common Member Selector, including Driver Selections, Assignments, Data Entry, and Traceability. To search and filter aliases, you must first select **Show Alias** from the Context Menu.

**Note:** The Alias View is not available on the Select Driver selector, which is accessed from Driver Definitions screen or from Driver Selections screen when adding or modifying a driver selection.

Aliases may be cloned if a particular dimension is cloned in Essbase.

To viewAliases:

1. From the application, select any screen that uses the Common Member Selector, for example, Driver Selections, Assignments, Data Entry, or Trace Allocations. The appropriate screen with Common Member Selector is displayed, showing all available members.

2. In the Common Member Selector, click the Context Menu button and select **Show Alias**.
   
   Aliases are displayed in the list of members.

   **Note:** If you select Show Alias from the Context Menu, and no alias is assigned, the member name is displayed within square brackets. For example, the member name Product is displayed as [Product]

**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 reuse 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 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 Dimension HierarchyType to “Enabled”, the first Gen2 member HierarchyType to “Stored” and the Gen2 member with alternative hierarchy with shared members to “Dynamic”.
  
  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 need not be created for ADS or flat files, because it is added automatically by Performance Management Architect. Do not delete this member.

**POV Dimensions**

POV dimensions indicate a specific point of view or version of the model, such as year, scenario, or period. The dimension can be customized to reflect the requirements of your organization. For example, the POV may consist of quarters, months, seasonal groupings, and so on.

At least one POV dimension is required for each model, but you can create up to four POV dimensions. The POV dimensions are set in the Performance Management Architect Property grid for the model.

A Version dimension is also available, and is used to create another instance of your model. This version can be modified to enable you to experiment with strategies or business options to play
“what-if” scenarios. By modifying the version, you can implement features, and compare results to determine the best course of action.

**Attribute Dimensions**

Attribute dimensions are a special type of dimension that are associated with a business dimension, and contain members that can be used to classify members of another, associated dimension. Attribute dimensions describe characteristics of data, such as the size and 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 assignment rules are created.

*Note:* For naming restrictions, see Appendix F, “Essbase Naming Conventions”.

Two Attribute dimension types exist in Performance Management Architect:

- **Attribute Dimensions:**
  - The attribute can be created using different structures, such as Boolean, Date, Numeric, and Text.
  - An attribute has a hierarchy, and the hierarchies can be rolled up for aggregate values.
  - Only one attribute from a given attribute dimension can be associated with one member.

- **User-Defined Attribute Dimensions (UDAs):**
  - The attribute can only be created using Text.
  - A UDA does not have a hierarchy and cannot be easily used in reports to generate sums.
  - Multiple UDAs can be associated with one member.

Each type of attribute dimension offers different advantages, depending on your model and reporting needs. For detailed information about working with attribute dimensions, see the *Oracle Essbase Database Administrator’s Guide*.

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.

For naming restrictions, see Appendix F, “Essbase Naming Conventions”.

**Dimension Sort Order and Density Settings**

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.
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 56.

The Dimension Sort Order settings for the model are validated in Performance Management Architect. See “Modifying the Standard Profitability Application Settings” on page 73.

You can also set the density for dimensions from the Dimension Sort Order Settings dialog box.

**Dimension Sort Order Recommendations**

Oracle recommends that you set the dimension sort using the following recommendations:

- 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, unique, and greater than or equal to 1.
- Measures 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 Recommendations” on page 56.

There are two ways to set the dimension sort order in Performance Management Architect:

- Using the Dimension Performance Settings dialog box to set the sort order for all dimensions at one time
- Using the Property Grid in the Dimension Library to set individual dimension sort order for one dimension at a time

> To set the Dimension Sort Order using the Dimension Performance Settings dialog box:

1. From EPM Workspace, select Navigate, then Administer, and then Dimension Library to display the Shared Library.
2 Right-click the name of the application, and select **Dimension Performance Settings**.

3 Select a dimension, and use the Up and Down arrows to move each dimension into the correct sort order, as described in “Setting the Dimension Sort Order” on page 56. The number of the original position for the dimension is displayed under Sort Order.

4 **Optional:** Under **Density** for the selected dimension, double-click the cell to display the options, and then select the appropriate density for the dimension. See “Optimizing Dimension Settings for Essbase” on page 60.

5 Click **OK**.

To set Dimension Sort Order for individual dimensions:

1 From EPM Workspace, select **Navigate**, then **Administer**, and then **Dimension Library**.

2 In the Profitability and Cost Management application, select the dimension for which you want to set the Dimension Sort Order.

3 In the Property Grid, select the Dimension Sort Order property, and under **Value**, enter the required number for the sort order. See “Setting the Dimension Sort Order” on page 56.

4 Click the Save icon.
Managing Detailed Profitability Applications and Dimensions using Performance Management Architect

In This Chapter

Working with Applications and Dimensions ........................................................... 123
Optimizing Detailed Profitability Dimension Settings for Essbase ................................... 124
Importing Detailed Profitability Metadata ........................................................................ 125
Detailed Profitability and Cost Management Dimension and Member Properties ................. 126
Creating Detailed Profitability Applications ...................................................................... 129
Validating and Deploying Detailed Profitability Applications in Performance Management Architect ............................................................................................................. 138

Working with Applications and Dimensions

Profitability and Cost Management uses Performance Management Architect or Profitability Applications 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.

Note: Lifecycle Management can be used not only to import or export model data, but also to import or export applications in Performance Management Architect.

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, 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:

- “Optimizing Detailed Profitability Dimension Settings for Essbase” on page 124
- “Importing Detailed Profitability Metadata” on page 125
- “Detailed Profitability and Cost Management Dimension and Member Properties” on page 126
- “Creating Detailed Profitability Applications” on page 129
- “Validating and Deploying Detailed Profitability Applications in Performance Management Architect” on page 138

### Optimizing Detailed Profitability Dimension Settings for Essbase

A typical Profitability and Cost Management application contains one MeasuresDetailed 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 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. In that case, the AllocationType and Measures dimension should be set to Sparse.

**Note:** The block size should still remain within the limits of the Essbase best practices recommendations. See 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. Set the associated Measures and AllocationType dimensions to Sparse. The decision 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 Detailed Profitability 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.

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 Enterprise Performance Management System User Security Administration Guide*.

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 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.

From the Dimension Library, you can edit the properties of these:

- **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 Administration Guide for detailed instructions on working with dimensions and members.

Detailed Profitability and Cost Management Dimension and Member Properties

The properties for Detailed Profitability and Cost Management dimensions and members are displayed in alphabetical order in Table 29, “Detailed Profitability and Cost Management Dimension and Member Properties”, which displays the following information:

- The Property Label, which provides a more readable display name for the property. If applicable, the associated database type is appended to the name (ASO). If no database 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 Administration 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 29 Detailed Profitability and Cost Management Dimension and Member Properties

<table>
<thead>
<tr>
<th>Property Label</th>
<th>Description</th>
<th>Property Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>Enter the alias for the selected dimension member. The alias is the alternate dimension member name that is displayed in a deployed application. To assign or change the default alias name for the current dimension member, click the existing alias name and enter the new one. Follow the proper naming rules. All databases have an alias table named Default.</td>
<td></td>
</tr>
<tr>
<td>Property Label</td>
<td>Description</td>
<td>Property Name</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------</td>
<td>---------------</td>
</tr>
</tbody>
</table>
| Application Type | Select the Application Type:  
  - General for Standard Profitability  
  - Detail for Detailed Profitability | ApplicationType |
| Attributes | Enter the characteristics of a dimension member, such as color, size, model, and so on.  
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 | Associate an attribute type with the selected member. Create attribute values for attributes assigned to dimension members that you can use to query on and to filter members.  
Only Text types are currently supported. | 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.  
You cannot assign a comment to an attribute dimension or member.  
**Note:** For Unicode-enabled databases, a maximum of 80 characters is allowed. | Comment |
| Consolidation | Member consolidation properties determine how children roll up into their parents.  
If the current member is not a dimension or an attribute, select the consolidation operator to assign to the member:  
- `+` (addition) - Default  
- `-` (subtraction)  
- `*` (multiplication)  
- `/` (division)  
- `%` (percent)  
- `~` (ignore during consolidation)  
- `^` (never consolidate)  
- NotUsed | Consolidation |
| Data Storage(ASO) | Select the ASO storage options for dimensions and members:  
- StoreData—Data is stored with the dimension.  
- 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.  
**Note:** This option does not apply to stored hierarchies in aggregate storage outlines.  
- LabelOnly—No data is associated with this dimension.  
- ShareData—Data associated with this member can be shared. The ShareData property applies to the member only. The Dimension Root Member cannot be shared. | ASODimensionDataStorage (for dimension root member)  
ASOMemberDataStorage (for dimension members) |
<table>
<thead>
<tr>
<th>Property Label</th>
<th>Description</th>
<th>Property Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimension Formula (ASO)</td>
<td>Double-click the cell, and then click the Selector button to open the Memo Editor. Enter the formula, and then click OK. Use the appropriate ASO MDX syntax.</td>
<td>ASODimensionFormula (for dimension root member)</td>
</tr>
<tr>
<td>Dimension Solve Order</td>
<td>Enter the numeric value in the solution sequence for selected dimensions. For example, if this dimension is to be solved second, enter “2.”</td>
<td>DimensionSolveOrder</td>
</tr>
<tr>
<td>Dimension Sort Order</td>
<td>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. For example, if this dimension is to be the second in the Essbase outline, enter “2.” Dimension Sort Order must be set for every dimension in the model, except Alias and UDA dimensions. The dimension sort order must be sequential, unique, and greater than or equal to 1. Set the sort order as outlined in “Setting the Dimension Sort Order” on page 56.</td>
<td>DimensionSortOrder</td>
</tr>
<tr>
<td>Hierarchy Type (Dimensions Only)</td>
<td>The <strong>Hierarchy Type</strong> property applies only to Aggregate Storage (ASO) databases in Essbase. For Detailed Profitability, this value is set to “Disabled.”</td>
<td>DimensionHierarchyType</td>
</tr>
<tr>
<td>Hierarchy Type (Members Only)</td>
<td>For Detailed Profitability, this value is set to “Disabled.”</td>
<td>HierarchyType</td>
</tr>
<tr>
<td>Level Usage for Aggregation</td>
<td>This property applies to the Gen1 or Gen2 hierarchy members who host a stored hierarchy in an ASO outline. For Detailed Profitability, this value is set to <strong>DEFAULT</strong></td>
<td>ASOMemberFormula</td>
</tr>
<tr>
<td>Member Formula (ASO)</td>
<td>Double-click the cell and then click the Selector button to open the Memo Editor. Enter the formula, and then click OK. Use the appropriate ASO MDX syntax.</td>
<td>MemberSolveOrder</td>
</tr>
<tr>
<td>Member Solve Order (Members Only)</td>
<td>Enter the numeric value in the solution sequence for selected member. For example, if this member is to be solved second, enter “2.” This property applies to ASO databases only. Members that have a solve order of 0 inherit the solve order of their dimension. Members with the same solve order are evaluated in the order in which their dimensions appear in the database outline, unless otherwise specified by the dimension sort order property. Members with no solve order are evaluated after members with a solve order.</td>
<td>IsPOVDimension</td>
</tr>
<tr>
<td>POV Dimension</td>
<td>Click the checkbox to set this dimension as a POV dimension for the selected model.</td>
<td>POVDimension</td>
</tr>
<tr>
<td>POV Display Order</td>
<td>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.</td>
<td>POVDisplayOrder</td>
</tr>
</tbody>
</table>
### 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.

Select one of the following options:

- **Default** – The view selection engine is free to decide which levels to pick. This is the initial value for new hierarchies.
- **AllLevels** – The view selection engine considers all levels of the hierarchy for aggregation.
- **NoAggregation** – The view selection engine cannot select any levels for aggregation.
- **TopLevelOnly** – The view selection engine considers only the highest level of the dimension for aggregation.
- **BottomaTop** – The view selection engine considers only the highest level and the lowest level of the dimension for aggregation.
- **BottomLevelOnly** – The view selection engine considers only the bottom level of the dimension for aggregation.

### UDA

Select the UDA member 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.

You can select UDA member of up to 80 alphanumeric characters.

*Note:* For Unicode-enabled databases, a maximum of 80 characters is allowed.

### Unicode

Select **Unicode** at the application level to set the application to Unicode mode. Unicode-mode applications support multiple character sets.

---

**Creating Detailed Profitability Applications**

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

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

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

- “Selecting the Detailed Profitability Application Type” on page 130
- “Selecting Detailed Profitability Dimensions” on page 130
- “Modifying the Application Settings” on page 135
Selecting the Detailed Profitability 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.

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

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 special characters, including “&” (ampersands). See Appendix F, “Essbase Naming Conventions”.

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

6. **Optional**: 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 Detailed Profitability Dimensions Manually” on page 134.

8. **Optional**: To create the dimensions, select **Auto Create Local Dimensions**.

   Selecting Auto Create Local Dimensions automatically creates MeasuresDetailed dimensions.

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

9. **Under Profitability**, select **Create as Detailed Application**.

10. **Click Next**. See “Selecting Detailed Profitability Dimensions” on page 130.

Selecting Detailed Profitability 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:

- MeasuresDetailed is a reserved dimension that is required for calculation and stage balancing. It contains AllocationMeasures but does not include Driver Measures.

   **Caution!** Do not edit this dimension, as modifications may result in the loss of data or the corruption of your model.
• At least one Point of View dimension (POV) must be defined by the user. The model POV provides a specific view of your model information for a selected POV or period. For example, POV dimension types may include some of the following dimensions:
  o Year
  o Time
  o Period
  o Scenario
  o Months
  o Financial Quarters
  o Seasons

• At least one Business Dimension must be defined by the user. Business dimensions are created to describe the business elements within the model, such as business-specific departments, general ledger accounts, activities, locations, customer, or products.

Business dimensions may include some or all of the following dimension types, and may apply to one or more stages or models:
  o Generic
  o Account
  o Entity
  o Time
  o 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 or color of products.

There are two Attribute dimension types in Performance Management Architect:
  o Attribute dimensions:
    - The attribute are created using Text.
    - An attribute has a hierarchy, and the hierarchies can be rolled up for aggregate values.
    - Only one attribute can be associated with one member.
  o User-defined attribute dimensions (UDAs):
    - The attribute can only be created using Text.
    - A UDA does not have a hierarchy, and cannot be easily used in reports to generate sums.
    - Multiple UDAs can be associated with a member.
For detailed information about working with attribute dimensions, see *Oracle Essbase Database Administrator’s Guide*.

- Alias is optional, and only required if you want to use aliases in your model. For naming conventions, see Appendix F, “Essbase Naming Conventions”.

**Caution!** Duplicate member names or aliases are not allowed within the same dimension.

**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 Detailed Profitability Dimensions Using the Wizard” on page 132.
- If you created a blank application, see “Adding Detailed Profitability Dimensions Manually” on page 134.

**Adding Detailed Profitability 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:

- MeasuresDetailed Dimension
- POV Dimension
- Alias Dimension (Optional)
- Business Dimensions
- Attribute Dimensions (Optional)

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).

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 view restricted words and characters for names, see Appendix F, “Essbase Naming Conventions”.

To define dimensions:

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

**Caution!** This dimension is required for Detailed Profitability applications, but it must NOT be modified or your application may be corrupted.

3 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.

4 Select a dimension from the drop-down list.

5 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**.
6 Repeat step 2 to step 4 to select or create new dimensions.

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

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

8 Click **Next**. The Application Settings screen is displayed.

9 Modify the application settings, as required. See “Modifying the Application Settings” on page 135.

---

### Adding Detailed Profitability 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**—Displays 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 Administration 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.**

   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 137.
6 Modify the application settings, as required. See “Modifying the Application Settings” on page 135.

Modifying the Application Settings

Subtopics

- Modifying Application Settings Using the Dimension Library
- Modifying Application Settings Using the Application Wizard
- Setting Properties for POV Dimensions
- Reordering Children Members

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 135.
- Use the Application Wizard for auto-generated applications. See “Modifying Application Settings Using the Application Wizard” on page 136.

Modifying Application Settings Using the Dimension Library

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

- For a Profitability and Cost Management model, perform the following tasks:
  - Set the Properties for POV Dimensions, and the POV Display Order for multiple POV dimensions, if required.
    See “Setting Properties for POV Dimensions” on page 136.
  - Set the Dimension Sort Order for all model dimensions.
    See “Dimension Sort Order and Density Settings” on page 120
  - Reorder the NoMember to display this member as the last generation 2 member on the list.
    See “Reordering Children Members” on page 137
• Modify dimensions, as required, using the procedures provided in the Oracle Hyperion Enterprise Performance Management Architect Administration Guide:
  o Set the dimension alias.
  o View application membership.
  o View dimension associations.
  o Synchronize dimensions.
  o Create members.
  o Insert shared members.
  o Manage orphan members.

Modifying Application 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.

Use the right-click menu to modify the application dimensions, and perform these tasks:

• For a Profitability and Cost Management model, perform the following tasks:
  o Set the Properties for POV Dimensions, and the POV Display Order for multiple POV dimensions, if required.
    See “Setting Properties for POV Dimensions” on page 136.
  o Set the Dimension Sort Order for all model dimensions.
    See “Dimension Sort Order and Density Settings” on page 120
  o Reorder the NoMember to display this member as the last generation 2 member on the list.
    See “Reordering Children Members” on page 137

• Modify dimensions, as required, using the procedures provided in the Oracle Hyperion Enterprise Performance Management Architect Administration Guide:

Setting Properties for POV Dimensions

At least one POV, or Point of View, dimension must be set for every Detailed 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 In the Dimension Library, 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 Gen2 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. Select the dimension to be reordered, as described below:
   - For Shared dimensions, under **Shared Library**, right-click the dimension that you want to reorder and then select **Reorder Children**.
   - For Local dimensions, under the Application column, right-click the dimension that you want to reorder and then select **Reorder Children**.

The Reorder Children dialog box is displayed.
3 Under Sort Order, double-click the cell for which the member is to be reordered, and enter the number of the new sort order location.

![Reorder Children window]

Note: Ensure the NoMember dimension is moved to the end of the list as the last Gen2 member; otherwise, the deployment will fail.

4 Repeat step 3 to order all members.

5 To preview the changes, click Preview.

6 Click Save.

Validating and Deploying Detailed Profitability Applications in Performance Management Architect

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

See these sections:
- “Detailed Profitability and Cost Management Validations” on page 139
- “Validating and Deploying Detailed Profitability Applications Using the Wizard” on page 140
- “Validating and Deploying Detailed Profitability Applications Using Performance Management Architect” on page 141

Note: You can add, rename, reparent, or delete individual members, multiple members, or combinations of members during redeployment of an application.

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

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

<table>
<thead>
<tr>
<th>Validation Level</th>
<th>Validations</th>
</tr>
</thead>
</table>
| Application      | - The name of the application must be seven characters or fewer, and cannot contain special characters.  
                  - At least one dimension must be set to POV type. Up to four dimensions may be marked as POV dimensions. Only one occurrence of each POV dimension class is allowed.  
                  - At least one business dimension must be defined.  
                  - At least one MeasuresDetailed dimension must be defined.  
                  - There is only one dimension of type “Account.”  
                  - There is only one dimension of type “Entity.”  
                  - Application names do not include Essbase special characters and reserved words. |
| Dimension        | - Root members of business dimensions must have the ASO data storage set to **LabelOnly**.  
                  - Dimension Sort Order is set for all dimensions in the model, excluding Alias and UDA dimensions, and satisfies the following conditions:  
                    - A dimension sort order must be set for every dimension in the model, except Alias and UDA dimensions.  
                    - Note: The Alias and UDA dimensions are ignored for Dimension Sort Order.  
                    - The dimension sort order must be sequential.  
                    - Measures dimension is set to 1 by default.  
                    - POV and business dimensions are set to 3 or higher.  
                    - Attribute dimensions are 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.  
                  - Business dimensions should have at least two Gen 2 members.  
                  - POV dimensions must have at least one member.  
                  - Attribute dimensions must satisfy the following conditions:  
                    - Only a Level-0 member from the attribute dimension may be assigned as an attribute.  
                    - Attributes may be assigned only to members at the same level in the base dimension.  
                    - Attribute dimensions can only be associated with sparse dimensions. |
Validating and Deploying Detailed Profitability Applications Using the Wizard

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.

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 “Validating and Deploying Detailed Profitability Applications Using Performance Management Architect” on page 141.
Validating and Deploying Detailed Profitability Applications Using Performance Management Architect

To validate and deploy a Detailed Profitability application using Performance Management Architect:

1. From EPM Workspace, select **Navigate**, then **Administer**, and then **Application Library** to display the Application Library.

2. 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 “Detailed Profitability and Cost Management Validations” on page 139.

   When the validation is complete, the Job Task dialog box is displayed, showing the associated job number.

3. Click the link for the associated job to view the Job Console.

4. 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.

5. Fix any errors listed in the Validation Log.

6. Repeat step 2 to step 5 until the validation succeeds with no errors.

   The application is ready to be deployed.

7. In the Application Library, right-click the application name, and select **Deploy** from the drop-down list.

   Applications are validated before they are deployed. Any validation errors display in the Job Console. You must correct all validation errors before deploying. When the validation is complete, the Deploy dialog box is displayed.

8. Provide the required information for the deployment:

   - **Optional:** Under **Notes**, add notes pertaining to the deployment of this application. These notes are appended to the job description in the Job Console. This text field allows up to 200 characters and may contain any characters.

   - **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 to which this application is to be assigned.

   - **Optional:** Decide whether **Full Deploy** should be selected, as follows:

     - If this is the first deployment of the application, the **Full Deploy** checkbox is disabled because Performance Management Architect automatically sends all dimensions to Profitability and Cost Management.
If Full Deploy is checked for any subsequent deployment, then Performance Management Architect sends all dimensions to Profitability and Cost Management.

If Full Deploy is not checked, then Performance Management Architect sends only modified dimensions to Profitability and Cost Management. If no changes were made to a dimension, then that dimension is not included in the deployment.

9 Click Deploy.

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

10 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.

**Note:** If the deployment of the application is not successfully completed, from the Application Library, right-click the application name, and select Application Cleanup. This option forces a delete call to both Shared Services and Profitability and Cost Management to remove any traces of the application, and resets the application in Performance Management Architect to the undeployed state.

**Caution!** There is a new dependency for the Detailed Profitability calculation engine on the dimension hierarchy reporting tables that were previously optional. These tables are deployed from the Calculate, Manage Database menu, Reporting Tables and Views tab within Profitability and Cost Management. You must redeploy these tables each time you redeploy a Detailed Profitability application from Enterprise Performance Management Architect (EPMA). For instructions, see “Deploying Detailed Profitability Reporting Views” in the Oracle Hyperion Profitability and Cost Management User’s Guide.
Calculating Detailed Profitability Models

In This Chapter

- Pre- and Post-Calculation Custom Scripts ........................................... 143
- Creating the ODBC Data Source to Enable Data Transfers .................. 146
- Advanced Calculation Options .............................................................. 146
- Other Process Type .............................................................................. 147

After validating the model, you can calculate the model.

From the Manage Calculation screen, the Detailed Profitability user selects the processing options, including Custom Scripts or specific Data POVs.

In addition to the basic calculation functions, Administrative users can also view additional operation and process types.

Caution!

There is a new dependency for the Detailed Profitability calculation engine on the dimension hierarchy reporting tables that were previously optional. These tables are deployed from the Calculate, Manage Database menu, Reporting Tables and Views tab within Profitability and Cost Management. You must redeploy these tables each time you redeploy a Detailed Profitability application from Enterprise Performance Management Architect (EPMA). For instructions, see “Deploying Detailed Profitability Reporting Views” in the Oracle Hyperion Profitability and Cost Management User’s Guide.

Pre- and Post-Calculation Custom Scripts

You can create custom Pre- and Post-Calculation scripts to perform tasks either before or after calculation of the application to manipulate the Model or Stage data.

The custom scripts are run, as follows:

- Pre-calculation SQL scripts run before the Model calculations.
- Post-calculation SQL scripts run after the Model calculations.

The custom scripts are run to prepare for allocations before the Model calculation or to enhance the results for reporting after Model calculations.
Using SQL Developer or any third-party SQL tool, you create a simple series of SQL statements, separated by a delimiter. The custom scripts are stored in the SCRIPT column of the HPM_SQL_SCRIPT table in the database, within the Product Schema.

Select the custom script when you are creating your Calculation Parameters. Under Task Areas, select Calculate, then Manage Calculation, and then select the required pre- or post-calculation script. For detailed instructions, see the Oracle Hyperion Profitability and Cost Management User’s Guide.

See these sections:

- “HPM_SQL_SCRIPT” on page 144
- “Creating Custom Scripts” on page 145

### HPM_SQL_SCRIPT

This table stores your custom pre- and post-calculation scripts. Each row with the table contains one or more SQL statements that can be executed during a stage model.

**Note:** Do not enter your script directly into the HPM_SQL_SCRIPT. See “Creating Custom Scripts” on page 145.

<table>
<thead>
<tr>
<th>Column</th>
<th>Data Type</th>
<th>Nullable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME</td>
<td>Varchar2 (80)</td>
<td>No</td>
<td>Name of the custom script</td>
</tr>
<tr>
<td>APPLICATION_NAME</td>
<td>Varchar2 (8)</td>
<td>No</td>
<td>Name of the application for which this custom script has been created</td>
</tr>
<tr>
<td>MODEL_POV_NAME</td>
<td>Varchar2 (80)</td>
<td>Yes</td>
<td>The selected POV is displayed in the format, &quot;Year&quot;:&quot;2012&quot;, &quot;Period&quot;:&quot;January&quot;, &quot;Scenario&quot;:&quot;Actual&quot;.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> If this column is set to NULL, the script can be used for any Model POV. If a specific POV is selected, the script can only be used for that POV.</td>
</tr>
<tr>
<td>SCRIPT_TYPE</td>
<td>Varchar2 (80)</td>
<td>Yes</td>
<td>Select the appropriate script type:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- PRE (displays Pre-Script)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- POST (displays Post-Script)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- EITHER of (displays Either Pre-Script or Post-Script)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- If no script type is specified, EITHER is assumed.</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>Varchar2 (255)</td>
<td>Yes</td>
<td>Enter a description of the purpose or contents of the script</td>
</tr>
<tr>
<td>SCRIPT</td>
<td>LONG</td>
<td>No</td>
<td>Enter the custom script here.</td>
</tr>
</tbody>
</table>

See “Creating Custom Scripts” on page 145 for instructions on creating the script.
Creating Custom Scripts

The custom scripts can be created in a text editor, such as Notepad, Textpad, or Oracle SQL Developer. The script may be as short as a single SQL statement, but if it is longer, a delimiter is required between statements. The required delimiter is the forward slash character ("/") on a separate line by itself. Note that stored procedures may be executed in these scripts.

The script is pasted or otherwise loaded into the SCRIPT column of the HPM_SQL_SCRIPT table, as shown in “HPM_SQL_SCRIPT” on page 144. Do not type the script directly into the HPM_SQL_SCRIPT table.

Caution! Only the system administrator who has the password to the HPM Product schema can create pre- and post-scripts.

➢ To create and load a custom script:

1. Create your custom script within a text editor, such as Notepad, Textpad, or Oracle SQL Developer.
2. Using Oracle SQL Developer, open the HPM_SQL_SCRIPT table. The table should be in the HPM Product Schema, in the same location as the product tables.
3. Insert a new row.
4. Enter the required values in the table, including the script type. See “HPM_SQL_SCRIPT” on page 144.
5. Copy the script into the SCRIPT column, as follows:
   - For Oracle using SQL Developer, copy and paste the script from a text editor, or use an update statement to populate the SCRIPT column. Here is an example of a SQL script for Oracle:
     ```
     update my_table set my_column =5
     /
     update your_table set your_column =6
     /
     begin myproc; end;
     /
     ```
   - For Microsoft SQL Server using SQL Studio, use an Update statement to populate the SCRIPT column. Here is an example of a SQL script for SQL Server:
     ```
     update my_table set my_column =5
     /
     update your_table set your_column =6
     /
     begin execute my_proc end
     /
     ```

Caution! You must include the delimiter “/” between multiple SQL statements in the same script. The “/” character must be on a line by itself at the end of each statement. Be sure the final “/” is followed by a newline character.
Creating the ODBC Data Source to Enable Data Transfers

The Data Transfers option in the Manage Calculation tab requires an ODBC data source called **PROFITABILITY_DS** that connects to the Product Schema in the Oracle or Microsoft SQL Server database that supports your Profitability and Cost Management instance.

To create this ODBC data source:

1. On the application server, open the ODBC administration utility: on a command line, execute the `odbcad32` command.
2. Select the **System DSN** tab and click **Add**....
3. Select the appropriate ODBC driver and click **Next**.
4. In **Data Source Name**, enter **PROFITABILITY_DS** and then fill the connection details to connect to the Product Schema.

Advanced Calculation Options

On the Manage Calculation screen, the Driver Operation Types and Other Process Type tabs are displayed for Administrators that are not available to basic users:

- “Driver Operation Types” on page 146
- “Other Process Type” on page 147

**Caution!** Oracle recommends that only experienced Database Administrators use these options, Modifications of any type to the Driver Operation Types or Other Process Types may have a profound impact on your applications, and may corrupt your model or data.

Driver Operation Types

The Driver Operation Types tab on the Manage Calculation screen displays the current driver operation types for Detailed Profitability.

**Caution!** Do not create new driver operation types on this tab. Only an experienced Database Administrator should access this tab, as any changes may have a profound impact on your applications, and may corrupt your model or data.

The Driver Operation Types is an administration feature intended for extending supported driver operations to solve performance issues or unique driver challenges. Its use requires advanced SQL knowledge, advanced understanding of Database Administration, and advanced understanding of an undocumented feature of Profitability and Cost Management Detailed
Profitability applications. Oracle does not recommend the use of this feature unless requested by Oracle in response to a customer service request.

**Other Process Type**

Two alternative calculation processes types are available for this release of Detailed Profitability:

- Oracle Database 11g (the default)
- Oracle Database 10g

If your system is configured with Oracle Database 10g, use the Other Process Type tab to select the appropriate calculation process for your database.

**Caution!** Oracle recommends that only experienced Database Administrators use the Other Process Type tab. DO NOT make any other modifications to the Other Process Type, as they may have a profound impact on your applications, and may corrupt your model or data.

Depending on the Oracle database version that you are using for Detailed Profitability, use this screen to change your database to the correct version.

➢ To change your Oracle Database version:

1. From EPM Workspace, select **Navigate**, then **Application**, then **Profitability** and then select the application for which you want to modify the Oracle Database version.
2. Under Task Areas, select **Calculate**, then **Manage Calculation**, and then the **Other Process Types** tab.

   **Note:** This tab is only visible if you have been assigned an Administrator user role.

3. **Select Dimension Group Synchronize**, and then **click the Edit button**. The Define Other Process Type dialog box is displayed.
4 Under Calculation Process, select the appropriate option for your Oracle database:
   - Dimension Group Synchronize (Oracle 11g)
   - Dimension Group Synchronize 10G (Oracle 10g)

5 Click **OK** to save the change.
In order to import model data from relational databases into Profitability and Cost Management, you must create a set of staging tables in a database schema that is separate from the Profitability and Cost Management product schema. (You may re-use the same schema used as the Model Data Schema for Detailed Profitability). You then populate these staging tables with the details of the application artifacts to be imported.

**Caution!** Do not modify the product schema.

Staging database scripts are available for Microsoft SQL Server and Oracle Database. Use the appropriate script to create the import tables in a new database schema:

Use the procedure and schema tables in this appendix to create staging tables:

- “Creating Import Database Tables for Detailed Profitability” on page 150
- “HPM_STGD_POV” on page 150
- “HPM_STGD_DRIVER” on page 151
- “HPM_STGD_DRIVER_SEL” on page 152
- “HPM_STGD_DRIVER_EXCEP” on page 153
- “HPM_STGD_ASGN_RULE_SEL” on page 154
- “HPM_STGD_CALCRULE_SNGLSRC” on page 156
- “HPM_STGD_CALCRULE_CALCMSRS” on page 157
- “HPM_STGD_CALCRULE_MULTISRC” on page 159
Creating Import Database Tables for Detailed Profitability

Staging database scripts are available for Microsoft SQL Server and Oracle Database. Use the `create_dp_staging.sql` script to create the tables in Detailed Profitability.

To create staging tables:

1. Create a new Oracle or Microsoft SQL Server database schema, outside of the product schema.
2. Locate `create_dp_staging.sql` in the default location for your database type:
   - `%EPM_ORACLE_HOME%/products/Profitability/database/Common/MSSQLServer`
   - `%EPM_ORACLE_HOME%/products/Profitability/database/Common/Oracle`
3. Run the `create_dp_staging.sql` script.

HPM_STGD_POV

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

Table 32  HPM_STGD_POV

<table>
<thead>
<tr>
<th>Field</th>
<th>SQL Server Data Type</th>
<th>Oracle Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>identity</td>
<td>integer(38,0)</td>
<td>Yes</td>
<td>Unique record ID</td>
</tr>
<tr>
<td>pov_dim1_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes</td>
<td>Name of the first dimension member in the selected POV</td>
</tr>
<tr>
<td>pov_dim2_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Name of the second dimension member in the selected POV</td>
</tr>
<tr>
<td>pov_dim3_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Name of the third dimension member in the selected POV</td>
</tr>
<tr>
<td>pov_dim4_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Name of the fourth dimension member in the selected POV</td>
</tr>
<tr>
<td>pov_state</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes</td>
<td>The current status of the POV:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Draft</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Published</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Archived</td>
</tr>
<tr>
<td>last_upload_date</td>
<td>timestamp (6)</td>
<td>timestamp (6)</td>
<td></td>
<td>The date and time of the last import</td>
</tr>
<tr>
<td>exception</td>
<td>nvarchar (255)</td>
<td>varchar2(255 CHAR)</td>
<td></td>
<td>Message detailing errors that occurred during the import of this table.</td>
</tr>
<tr>
<td>Field</td>
<td>SQL Server Data Type</td>
<td>Oracle Data Type</td>
<td>Required</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>----------------------</td>
<td>---------------------</td>
<td>----------</td>
<td>------------------------------------------------------------------</td>
</tr>
<tr>
<td>created_userid</td>
<td>nvarchar (32)</td>
<td>varchar2(32 CHAR)</td>
<td>Populated by the import program upon update</td>
<td>The ID of the user who initiated the last import</td>
</tr>
<tr>
<td>created_timestamp</td>
<td>timestamp (6)</td>
<td>timestamp (6)</td>
<td></td>
<td>The date and time of the last import</td>
</tr>
<tr>
<td>modified_userid</td>
<td>nvarchar (32)</td>
<td>varchar2(32 CHAR)</td>
<td>Populated by the import program upon update</td>
<td>The ID of the user who modified the last import</td>
</tr>
<tr>
<td>modified_timestamp</td>
<td>timestamp (6)</td>
<td>timestamp (6)</td>
<td></td>
<td>The date and time of the modified import</td>
</tr>
</tbody>
</table>

**HPM_STGD_DRIVER**

The HPM_STGD_DRIVER table provides details about the driver, including driver type, display order and formula.

*Table 33  HPM_STGD_DRIVER*

<table>
<thead>
<tr>
<th>Field</th>
<th>SQL Server Data Type</th>
<th>Oracle Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>identity</td>
<td>integer(38,0)</td>
<td>Yes</td>
<td>Unique record ID</td>
</tr>
<tr>
<td>name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes</td>
<td>Name of the selected driver</td>
</tr>
<tr>
<td>description</td>
<td>nvarchar (255)</td>
<td>varchar2(255 CHAR)</td>
<td></td>
<td>Description of the purpose for the selected driver</td>
</tr>
<tr>
<td>formula</td>
<td>nvarchar (500)</td>
<td>varchar2(500 CHAR)</td>
<td>Yes</td>
<td>Formula created for the driver by a user. This formula must be created using SQL syntax, and mathematically correct operations.</td>
</tr>
</tbody>
</table>
| allow_idle_cost   | nvarchar (1)         | varchar2 (1 CHAR)   |          | Enter the appropriate value to determine whether idle costs are allowed for this driver:  
|                   |                      |                     |          |  ● Enter “Y” (Yes) to allow idle cost for a driver.  
|                   |                      |                     |          |  ● Enter “N” (No) to disable idle cost for a driver. This option is the default.  |
| priority          | integer(38,0)        | integer(38,0)       | Yes      | Enter the calculation priority of a driver so that allocations within a stage can be run in the specified order.  
|                   |                      |                     |          | The driver with the lowest priority is processed first. By default, the priority is set to 100, and the highest priority is 1. Numbers need not be sequential.  
<p>|                   |                      |                     |          | Drivers with the same priority are executed in no particular order. Only whole, positive numbers are valid.  |
| volume_formula    | nvarchar (4000)      | varchar2(4000 CHAR) |          | For Rate-Based drivers only, which formula has both the Rate and Volume formulas defined. |</p>
<table>
<thead>
<tr>
<th>Field</th>
<th>SQL Server Data Type</th>
<th>Oracle Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dst_measure_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes</td>
<td>Name of the measure on the Destination table</td>
</tr>
<tr>
<td>tdv_measure_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Measure that is used as the denominator in the allocation formula - DV/TDV</td>
</tr>
<tr>
<td>operation_type</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes</td>
<td>The operation type selected for the driver:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Rate-based Allocation</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Ratio-based Allocation</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Calculated Measures</td>
</tr>
<tr>
<td>last_upload_date</td>
<td>timestamp (6)</td>
<td>timestamp (6)</td>
<td></td>
<td>The date and time of the last import</td>
</tr>
<tr>
<td>exception</td>
<td>nvarchar (255)</td>
<td>varchar2(255 CHAR)</td>
<td></td>
<td>Message detailing errors that occurred during the import of this table.</td>
</tr>
<tr>
<td>created_userid</td>
<td>nvarchar (32)</td>
<td>varchar2(32 CHAR)</td>
<td></td>
<td>The ID of the user who initiated the last import</td>
</tr>
<tr>
<td>created_timestamp</td>
<td>timestamp (6)</td>
<td>timestamp (6)</td>
<td></td>
<td>The date and time of the last import</td>
</tr>
<tr>
<td>modified_userid</td>
<td>nvarchar (32)</td>
<td>varchar2(32 CHAR)</td>
<td></td>
<td>The ID of the user who modified the last import</td>
</tr>
<tr>
<td>modified_timestamp</td>
<td>timestamp (6)</td>
<td>timestamp (6)</td>
<td></td>
<td>The date and time of the modified import</td>
</tr>
</tbody>
</table>

**HPM_STGD_DRIVER_SEL**

The HPM_STGD_DRIVER_SEL table provides details about the POV and source stage driver dimension member for selected driver rules.

<table>
<thead>
<tr>
<th>Field</th>
<th>SQL Server Data Type</th>
<th>Oracle Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>identity</td>
<td>integer(38,0)</td>
<td>Yes</td>
<td>Unique record ID</td>
</tr>
<tr>
<td>pov_dim1_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes</td>
<td>Name of the first dimension member in the selected POV</td>
</tr>
<tr>
<td>pov_dim2_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Name of the second dimension member in the selected POV, if applicable</td>
</tr>
<tr>
<td>Field</td>
<td>SQL Server Data Type</td>
<td>Oracle Data Type</td>
<td>Required</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>----------------------</td>
<td>----------------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>pov_dim3_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Name of the third dimension member in the selected POV, if applicable</td>
</tr>
<tr>
<td>pov_dim4_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Name of the fourth dimension member in the selected POV, if applicable</td>
</tr>
<tr>
<td>driver_dim_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes</td>
<td>Name of the selected driver dimension member</td>
</tr>
<tr>
<td>driver_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes</td>
<td>Name of the selected driver</td>
</tr>
<tr>
<td>last_upload_date</td>
<td>timestamp (6)</td>
<td>timestamp (6)</td>
<td></td>
<td>The date and time of the last import</td>
</tr>
<tr>
<td>import_exception</td>
<td>nvarchar (255)</td>
<td>varchar2(255 CHAR)</td>
<td></td>
<td>Message detailing errors that occurred during the import of this table.</td>
</tr>
<tr>
<td>created_userid</td>
<td>nvarchar (32)</td>
<td>varchar2(32 CHAR)</td>
<td></td>
<td>Populated by the import program upon update</td>
</tr>
<tr>
<td>created_timestamp</td>
<td>timestamp (6)</td>
<td>timestamp (6)</td>
<td></td>
<td>The date and time of the last import</td>
</tr>
<tr>
<td>modified_userid</td>
<td>nvarchar (32)</td>
<td>varchar2(32 CHAR)</td>
<td></td>
<td>Populated by the import program upon update</td>
</tr>
<tr>
<td>modified_timestamp</td>
<td>timestamp (6)</td>
<td>timestamp (6)</td>
<td></td>
<td>The date and time of the modified import</td>
</tr>
</tbody>
</table>

**HPM_STGD_DRIVER_EXCEP**

The HPM_STGD_DRIVER_EXCEP table provides details about the POV and source stage intersection for the selected driver exceptions.

**Table 35  HPM_STGD_DRIVER_EXCEP**

<table>
<thead>
<tr>
<th>Field</th>
<th>SQL Server Data Type</th>
<th>Oracle Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>identity</td>
<td>integer(38,0)</td>
<td>Yes</td>
<td>Unique record ID</td>
</tr>
<tr>
<td>pov_dim1_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes</td>
<td>Name of the first dimension member in the selected POV</td>
</tr>
<tr>
<td>pov_dim2_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Name of the second dimension member in the selected POV, if applicable</td>
</tr>
<tr>
<td>pov_dim3_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Name of the third dimension member in the selected POV, if applicable</td>
</tr>
<tr>
<td>Field</td>
<td>SQL Server Data Type</td>
<td>Oracle Data Type</td>
<td>Required</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>----------------------</td>
<td>------------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>pov_dim4_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Name of the fourth dimension member in the selected POV, if applicable</td>
</tr>
<tr>
<td>src_dim1_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes</td>
<td>Name of the first dimension member in the source stage</td>
</tr>
<tr>
<td>src_dim2_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Name of the second dimension member in the source stage, if applicable</td>
</tr>
<tr>
<td>src_dim3_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Name of the third dimension member in the source stage, if applicable</td>
</tr>
<tr>
<td>src_dim4_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Name of the fourth dimension member in the source stage, if applicable</td>
</tr>
<tr>
<td>src_dim5_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Name of the fifth dimension member in the source stage, if applicable</td>
</tr>
<tr>
<td>driver_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes</td>
<td>Name of the selected driver</td>
</tr>
<tr>
<td>last_upload_date</td>
<td>timestamp (6)</td>
<td>timestamp (6)</td>
<td></td>
<td>The date and time of the last import</td>
</tr>
<tr>
<td>import_exception</td>
<td>nvarchar (255)</td>
<td>varchar2(255 CHAR)</td>
<td></td>
<td>Message detailing errors that occurred during the import of this table.</td>
</tr>
<tr>
<td>created_userid</td>
<td>nvarchar (32)</td>
<td>varchar2(32 CHAR)</td>
<td>Populated by the import program upon update</td>
<td>The ID of the user who initiated the last import</td>
</tr>
<tr>
<td>created_timestamp</td>
<td>timestamp (6)</td>
<td>timestamp (6)</td>
<td></td>
<td>The date and time of the last import</td>
</tr>
<tr>
<td>modified_userid</td>
<td>nvarchar (32)</td>
<td>varchar2(32 CHAR)</td>
<td>Populated by the import program upon update</td>
<td>The ID of the user who modified the last import</td>
</tr>
<tr>
<td>modified_timestamp</td>
<td>timestamp (6)</td>
<td>timestamp (6)</td>
<td></td>
<td>The date and time of the modified import</td>
</tr>
</tbody>
</table>

**HPM_STGD_ASGN_RULE_SEL**

The HPM_STGD_ASGN_RULE_SEL table provides details about the assignment rule selections.
<table>
<thead>
<tr>
<th>Field</th>
<th>SQL Server Data Type</th>
<th>Oracle Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>identity</td>
<td>integer(38,0)</td>
<td>Yes</td>
<td>Unique record ID</td>
</tr>
<tr>
<td>pov_dim1_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes</td>
<td>Name of the first dimension member in the selected POV</td>
</tr>
<tr>
<td>pov_dim2_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Name of the second dimension member in the selected POV</td>
</tr>
<tr>
<td>pov_dim3_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Name of the third dimension member in the selected POV</td>
</tr>
<tr>
<td>pov_dim4_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Name of the fourth dimension member in the selected POV</td>
</tr>
<tr>
<td>src_dim1_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes</td>
<td>Name of the first dimension member in the source stage</td>
</tr>
<tr>
<td>src_dim2_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Name of the second dimension member in the source stage, if applicable</td>
</tr>
<tr>
<td>src_dim3_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Name of the third dimension member in the source stage, if applicable</td>
</tr>
<tr>
<td>src_dim4_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Name of the fourth dimension member in the source stage, if applicable</td>
</tr>
<tr>
<td>src_dim5_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Name of the fifth dimension member in the source stage, if applicable</td>
</tr>
<tr>
<td>rule_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes</td>
<td>Name of the rule controlling the selected assignment. The rule must be present in the target database.</td>
</tr>
<tr>
<td>last_upload_date</td>
<td>timestamp (6)</td>
<td>timestamp (6)</td>
<td></td>
<td>The date and time of the last import</td>
</tr>
<tr>
<td>import_exception</td>
<td>nvarchar (255)</td>
<td>varchar2(255 CHAR)</td>
<td></td>
<td>Message detailing errors that occurred during the import of this table.</td>
</tr>
<tr>
<td>created_userid</td>
<td>nvarchar (32)</td>
<td>varchar2(32 CHAR)</td>
<td></td>
<td>Populated by the import program upon update</td>
</tr>
<tr>
<td>created_timestamp</td>
<td>timestamp (6)</td>
<td>timestamp (6)</td>
<td></td>
<td>The date and time of the last import</td>
</tr>
<tr>
<td>modified_userid</td>
<td>nvarchar (32)</td>
<td>varchar2(32 CHAR)</td>
<td></td>
<td>Populated by the import program upon update</td>
</tr>
<tr>
<td>Field</td>
<td>SQL Server Data Type</td>
<td>Oracle Data Type</td>
<td>Required</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>----------------------</td>
<td>------------------</td>
<td>----------</td>
<td>-------------------------------------------------</td>
</tr>
<tr>
<td>modified_timestamp</td>
<td>timestamp (6)</td>
<td>timestamp (6)</td>
<td></td>
<td>The date and time of the modified import</td>
</tr>
</tbody>
</table>

**HPM_STGD_CALCRULE_SNGLSRC**

The HPM_STGD_CALCRULE_SNGLSRC table provides details about creating or updating a single source calculation rule.

**Table 37  HPM_STGD_CALCRULE_SNGLSRC**

<table>
<thead>
<tr>
<th>Field</th>
<th>SQL Server Data Type</th>
<th>Oracle Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>identity</td>
<td>NUMBER(38,0)</td>
<td>Yes</td>
<td>Unique identifier for each row. This can come from any sequence generator.</td>
</tr>
<tr>
<td>name</td>
<td>nvarchar (80)</td>
<td>VARCHAR2(80 CHAR)</td>
<td>Yes</td>
<td>Name of the calculation rule.</td>
</tr>
<tr>
<td>description</td>
<td>nvarchar (255)</td>
<td>varchar2(255 CHAR)</td>
<td></td>
<td>Long description of the calculation rule.</td>
</tr>
<tr>
<td>pov_dim1_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes</td>
<td>Name of the first dimension member in the selected POV</td>
</tr>
<tr>
<td>pov_dim2_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Name of the second dimension member in the selected POV</td>
</tr>
<tr>
<td>pov_dim3_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Name of the third dimension member in the selected POV</td>
</tr>
<tr>
<td>pov_dim4_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Name of the fourth dimension member in the selected POV</td>
</tr>
<tr>
<td>calculation_sequence</td>
<td>identity</td>
<td>NUMBER(4,0)</td>
<td>Yes</td>
<td>Numeric value designating the relative order in which the calculation rule will be processed during calculation. This must be &gt; 0 or you will get a check constraint error during insert.</td>
</tr>
<tr>
<td>tag</td>
<td>nvarchar (255)</td>
<td>VARCHAR2(255)</td>
<td></td>
<td>Specify search tags for the calculation rule in this column.</td>
</tr>
<tr>
<td>enabled_flag</td>
<td>nvarchar (1)</td>
<td>VARCHAR2(1 CHAR)</td>
<td>Yes</td>
<td>Designates the first driver sequence priority in the range of drivers to be executed for this rule. This must be &gt; 0 or you will get a check constraint error during insert.</td>
</tr>
<tr>
<td>first_sequence_priority</td>
<td>identity</td>
<td>NUMBER(38,0)</td>
<td>Yes</td>
<td>Designates the last driver sequence priority in the range of drivers to be executed for this rule. This must be &gt;= FIRST_SEQUENCE_PRIORITY or you will get a check constraint error during insert.</td>
</tr>
<tr>
<td>last_sequence_priority</td>
<td>identity</td>
<td>NUMBER(38,0)</td>
<td>Yes</td>
<td>Designates the last driver sequence priority in the range of drivers to be executed for this rule. This must be &gt;= FIRST_SEQUENCE_PRIORITY or you will get a check constraint error during insert.</td>
</tr>
<tr>
<td>Field</td>
<td>SQL Server Data Type</td>
<td>Oracle Data Type</td>
<td>Required</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>----------------------</td>
<td>------------------------</td>
<td>----------</td>
<td>--------------------------------------------------------------</td>
</tr>
<tr>
<td>all_drivers_flag</td>
<td>nvarchar (1)</td>
<td>VARCHAR2(1 CHAR)</td>
<td>Yes</td>
<td>When Y ignore values in the First Sequence and Last Sequence priority fields during calculation.</td>
</tr>
<tr>
<td>last_upload_date</td>
<td>timestamp</td>
<td>timestamp</td>
<td></td>
<td>This column will be populated by the import showing the most recent date that the import was performed.</td>
</tr>
<tr>
<td>import_exception</td>
<td>nvarchar (255)</td>
<td>varchar2(255 CHAR)</td>
<td></td>
<td>Contains the error code indicating if there was a problem with the import of the row. Null means the import was successful.</td>
</tr>
<tr>
<td>created_userid</td>
<td>nvarchar (32)</td>
<td>varchar2(32 CHAR)</td>
<td></td>
<td>Identifies the user who originally created the row in this table. This is not used by HPCM.</td>
</tr>
<tr>
<td>created_timestamp</td>
<td>timestamp (6)</td>
<td>timestamp</td>
<td></td>
<td>Identifies the date/time on which the row was originally created. This is not used by HPCM.</td>
</tr>
<tr>
<td>modified_userid</td>
<td>nvarchar (32)</td>
<td>varchar2(32 CHAR)</td>
<td></td>
<td>Identifies the user who originally created the row in this table. This is not used by HPCM.</td>
</tr>
<tr>
<td>modified_timestamp</td>
<td>timestamp (6)</td>
<td>timestamp</td>
<td></td>
<td>Identifies the date/time on which the row was originally created. This is not used by HPCM.</td>
</tr>
</tbody>
</table>

### HPM_STGD_CALCRULE_CALCMSRS

The HPM_STGD_CALCRULE_CALCMSRS table provides details about creating or updating a calculated measures rule.

<table>
<thead>
<tr>
<th>Field</th>
<th>SQL Server Data Type</th>
<th>Oracle Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>identity</td>
<td>number(38,0)</td>
<td>Yes</td>
<td>Unique identifier for each row. This can come from any sequence generator.</td>
</tr>
<tr>
<td>name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes</td>
<td>Name of the calculation rule.</td>
</tr>
<tr>
<td>description</td>
<td>nvarchar (255)</td>
<td>varchar2(255 CHAR)</td>
<td></td>
<td>Long description of the calculation rule.</td>
</tr>
<tr>
<td>pov_dim1_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes</td>
<td>Name of the first dimension member in the selected POV</td>
</tr>
<tr>
<td>pov_dim2_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Name of the second dimension member in the selected POV</td>
</tr>
<tr>
<td>pov_dim3_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Name of the third dimension member in the selected POV</td>
</tr>
<tr>
<td>pov_dim4_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Name of the fourth dimension member in the selected POV</td>
</tr>
<tr>
<td>Field</td>
<td>SQL Server Data Type</td>
<td>Oracle Data Type</td>
<td>Required</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>----------------------</td>
<td>------------------</td>
<td>----------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>calculation_sequence</td>
<td>number</td>
<td>number(4,0)</td>
<td>Yes</td>
<td>Numeric value designating the relative order in which the calculation rule will be processed during calculation. This must be &gt; 0 or you will get a check constraint error during insert.</td>
</tr>
<tr>
<td>tag</td>
<td>nvarchar (255)</td>
<td>varchar2(255)</td>
<td></td>
<td>Specify search tags for the calculation rule in this column.</td>
</tr>
<tr>
<td>enabled_flag</td>
<td>nvarchar (1)</td>
<td>varchar2(1 CHAR)</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>driver_name1</td>
<td>nvarchar (1)</td>
<td>varchar2(1 CHAR)</td>
<td>Yes</td>
<td>Driver name to be associated with this calculation rule.</td>
</tr>
<tr>
<td>driver_name2</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Driver name to be associated with this calculation rule.</td>
</tr>
<tr>
<td>driver_name3</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Driver name to be associated with this calculation rule.</td>
</tr>
<tr>
<td>driver_name4</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Driver name to be associated with this calculation rule.</td>
</tr>
<tr>
<td>driver_name5</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Driver name to be associated with this calculation rule.</td>
</tr>
<tr>
<td>dest_assign_rule_name1</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes</td>
<td>Destination Assignment Rule to be associated with this calculation rule.</td>
</tr>
<tr>
<td>dest_assign_rule_name2</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Destination Assignment Rule to be associated with this calculation rule.</td>
</tr>
<tr>
<td>dest_assign_rule_name3</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Destination Assignment Rule to be associated with this calculation rule.</td>
</tr>
<tr>
<td>dest_assign_rule_name4</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Destination Assignment Rule to be associated with this calculation rule.</td>
</tr>
<tr>
<td>dest_assign_rule_name5</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Destination Assignment Rule to be associated with this calculation rule.</td>
</tr>
<tr>
<td>last_upload_date</td>
<td>timestamp (6)</td>
<td>timestamp</td>
<td></td>
<td>This column will be populated by the import showing the most recent date that the import was performed.</td>
</tr>
<tr>
<td>import_exception</td>
<td>nvarchar (255)</td>
<td>varchar2(255 CHAR)</td>
<td></td>
<td>Contains the error code indicating if there was a problem with the import of the row. Null means the import was successful.</td>
</tr>
<tr>
<td>created_userid</td>
<td>nvarchar (32)</td>
<td>varchar2(32 CHAR)</td>
<td></td>
<td>Identifies the user who originally created the row in this table. This is not used by HPCM.</td>
</tr>
<tr>
<td>created_timestamp</td>
<td>timestamp (6)</td>
<td>timestamp</td>
<td></td>
<td>Identifies the date/time on which the row was originally created. This is not used by HPCM.</td>
</tr>
</tbody>
</table>
### modified_userid
- **SQL Server Data Type**: nvarchar (32)
- **Oracle Data Type**: varchar2(32 CHAR)
- **Required**: Yes
- **Description**: Identifies the user who originally created the row in this table. This is not used by HPCM.

### modified_timestamp
- **SQL Server Data Type**: timestamp (6)
- **Oracle Data Type**: timestamp
- **Required**: Yes
- **Description**: Identifies the date/time on which the row was originally created. This is not used by HPCM.

### HPM_STGD_CALCRULE_MULTISRC

The HPM_STGD_CALCRULE_MULTISRC table provides details about creating or updating a multi source calculation rule.

**Table 39  HPM_STGD_CALCRULE_CALCMSRS**

<table>
<thead>
<tr>
<th>Field</th>
<th>SQL Server Data Type</th>
<th>Oracle Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>identity</td>
<td>number(38,0)</td>
<td>Yes</td>
<td>Unique identifier for each row. This can come from any sequence generator.</td>
</tr>
<tr>
<td>name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes</td>
<td>Name of the calculation rule.</td>
</tr>
<tr>
<td>description</td>
<td>nvarchar (255)</td>
<td>varchar2(255 CHAR)</td>
<td></td>
<td>Long description of the calculation rule.</td>
</tr>
<tr>
<td>pov_dim1_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes</td>
<td>Name of the first dimension member in the selected POV</td>
</tr>
<tr>
<td>pov_dim2_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Name of the second dimension member in the selected POV</td>
</tr>
<tr>
<td>pov_dim3_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Name of the third dimension member in the selected POV</td>
</tr>
<tr>
<td>pov_dim4_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Name of the fourth dimension member in the selected POV</td>
</tr>
<tr>
<td>calculation_sequence</td>
<td>number</td>
<td>number(4,0)</td>
<td>Yes</td>
<td>Numeric value designating the relative order in which the calculation rule will be processed during calculation. This must be &gt; 0 or you will get a check constraint error during insert.</td>
</tr>
<tr>
<td>tag</td>
<td>nvarchar (255)</td>
<td>varchar2(255)</td>
<td></td>
<td>Specify search tags for the calculation rule in this column.</td>
</tr>
<tr>
<td>enabled_flag</td>
<td>nvarchar (1)</td>
<td>varchar2(1 CHAR)</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>driver_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes</td>
<td>Driver name to be associated with this calculation rule.</td>
</tr>
<tr>
<td>src_assign_rule_name1</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes</td>
<td>Source assignment rule to be associated with this calculation rule.</td>
</tr>
<tr>
<td>Field</td>
<td>SQL Server Data Type</td>
<td>Oracle Data Type</td>
<td>Required</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>----------------------</td>
<td>-----------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>src_assign_rule_name2</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Source assignment rule to be associated with this calculation rule.</td>
</tr>
<tr>
<td>src_assign_rule_name3</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Source assignment rule to be associated with this calculation rule.</td>
</tr>
<tr>
<td>src_assign_rule_name4</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Source assignment rule to be associated with this calculation rule.</td>
</tr>
<tr>
<td>src_assign_rule_name5</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Source assignment rule to be associated with this calculation rule.</td>
</tr>
<tr>
<td>dest_assign_rule_name1</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Yes</td>
<td>Destination assignment rule to be associated with this calculation rule.</td>
</tr>
<tr>
<td>dest_assign_rule_name2</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Destination Assignment Rule to be associated with this calculation rule.</td>
</tr>
<tr>
<td>dest_assign_rule_name3</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Destination assignment rule to be associated with this calculation rule.</td>
</tr>
<tr>
<td>dest_assign_rule_name4</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Destination assignment rule to be associated with this calculation rule.</td>
</tr>
<tr>
<td>dest_assign_rule_name5</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td></td>
<td>Destination assignment rule to be associated with this calculation rule.</td>
</tr>
<tr>
<td>last_upload_date</td>
<td>timestamp (6)</td>
<td>timestamp</td>
<td></td>
<td>This column will be populated by the import showing the most recent date that the import was performed.</td>
</tr>
<tr>
<td>import_exception</td>
<td>nvarchar (255)</td>
<td>VARCHAR2(255 CHAR)</td>
<td></td>
<td>Contains the error code indicating if there was a problem with the import of the row. Null means the import was successfull.</td>
</tr>
<tr>
<td>created_userid</td>
<td>nvarchar (32)</td>
<td>varchar2(32 CHAR)</td>
<td></td>
<td>Identifies the user who originally created the row in this table. This is not used by HPCM.</td>
</tr>
<tr>
<td>created_timestamp</td>
<td>timestamp (6)</td>
<td>timestamp</td>
<td></td>
<td>Identifies the date/time on which the row was originally created. This is not used by HPCM.</td>
</tr>
<tr>
<td>modified_userid</td>
<td>nvarchar (32)</td>
<td>varchar2(32 CHAR)</td>
<td></td>
<td>Identifies the user who originally created the row in this table. This is not used by HPCM.</td>
</tr>
<tr>
<td>modified_timestamp</td>
<td>timestamp (6)</td>
<td>timestamp</td>
<td></td>
<td>Identifies the date/time on which the row was originally created. This is not used by HPCM.</td>
</tr>
</tbody>
</table>
After the model has been created, you can query the database to display the model artifacts as output in a database view.

An Administrator can create 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.

**HPM_EXPD_STAGE**

The HPM_EXPD_STAGE view retrieves the stage name and order, the application name, and the Source Stage and Destination Stage dimensions.

<table>
<thead>
<tr>
<th>Field</th>
<th>SQL Server Data Type</th>
<th>Oracle Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>application_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the selected application</td>
</tr>
<tr>
<td>id</td>
<td>identity</td>
<td>integer(38,0)</td>
<td>Stage ID that is used in the Profitability and Cost Management database</td>
</tr>
<tr>
<td>name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the selected Source stage</td>
</tr>
<tr>
<td>description</td>
<td>nvarchar (255)</td>
<td>varchar2(255 CHAR)</td>
<td>Description of the purpose for the selected stage</td>
</tr>
<tr>
<td>Field</td>
<td>SQL Server Data Type</td>
<td>Oracle Data Type</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>----------------------</td>
<td>--------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>stage_order</td>
<td>integer</td>
<td>number (38,0)</td>
<td>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.</td>
</tr>
<tr>
<td>stage_prefix</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Differentiating prefix for the stage name.</td>
</tr>
<tr>
<td>driver_dim_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the selected driver dimension</td>
</tr>
</tbody>
</table>
| ls_intra      | nvarchar (1)         | varchar2(1 CHAR)  | Flag determining whether instrastage assignment are allowed:  
|               |                      |                    | • “Y” (Yes) specifies that instrastage assignments within the model stage are allowed.  
<p>|               |                      |                    | • “N” (No) specifies that instrastage assignments within the model stage are not allowed.                                                   |
| dim1_name     | nvarchar (80)        | varchar2(80 CHAR) | Name of the first dimension in the stage                                                                                                                                 |
| dim2_name     | nvarchar (80)        | varchar2(80 CHAR) | Name of the second dimension included in the stage, if available.                                                                               |
| dim3_name     | nvarchar (80)        | varchar2(80 CHAR) | Name of the third dimension included in the stage, if available.                                                                               |
| dim4_name     | nvarchar (80)        | varchar2(80 CHAR) | Name of the fourth dimension included in the stage, if available.                                                                               |
| dim5_name     | nvarchar (80)        | varchar2(80 CHAR) | Name of the fifth dimension included in the stage, if available.                                                                               |
| dim6_name     | nvarchar (80)        | varchar2(80 CHAR) | Name of a dimension in the Destination stage                                                                                                                                 |
| dim7_name     | nvarchar (80)        | varchar2(80 CHAR) | Name of a dimension in the Destination stage                                                                                                                                 |
| dim8_name     | nvarchar (80)        | varchar2(80 CHAR) | Name of a dimension in the Destination stage                                                                                                                                 |
| dim9_name     | nvarchar (80)        | varchar2(80 CHAR) | Name of a dimension in the Destination stage                                                                                                                                 |
| dim10_name    | nvarchar (80)        | varchar2(80 CHAR) | Name of a dimension in the Destination stage                                                                                                                                 |
| dim11_name    | nvarchar (80)        | varchar2(80 CHAR) | Name of a dimension in the Destination stage                                                                                                                                 |
| dim12_name    | nvarchar (80)        | varchar2(80 CHAR) | Name of a dimension in the Destination stage                                                                                                                                 |
| dim13_name    | nvarchar (80)        | varchar2(80 CHAR) | Name of a dimension in the Destination stage                                                                                                                                 |
| dim14_name    | nvarchar (80)        | varchar2(80 CHAR) | Name of a dimension in the Destination stage                                                                                                                                 |
| dim15_name    | nvarchar (80)        | varchar2(80 CHAR) | Name of a dimension in the Destination stage                                                                                                                                 |
| dim16_name    | nvarchar (80)        | varchar2(80 CHAR) | Name of a dimension in the Destination stage                                                                                                                                 |
| dim17_name    | nvarchar (80)        | varchar2(80 CHAR) | Name of a dimension in the Destination stage                                                                                                                                 |
| dim18_name    | nvarchar (80)        | varchar2(80 CHAR) | Name of a dimension in the Destination stage                                                                                                                                 |
| dim19_name    | nvarchar (80)        | varchar2(80 CHAR) | Name of a dimension in the Destination stage                                                                                                                                 |
| dim20_name    | nvarchar (80)        | varchar2(80 CHAR) | Name of a dimension in the Destination stage                                                                                                                                 |</p>
<table>
<thead>
<tr>
<th>Field</th>
<th>SQL Server Data Type</th>
<th>Oracle Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dim21_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of a dimension in the Destination stage</td>
</tr>
<tr>
<td>dim22_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of a dimension in the Destination stage</td>
</tr>
<tr>
<td>dim23_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of a dimension in the Destination stage</td>
</tr>
<tr>
<td>dim24_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of a dimension in the Destination stage</td>
</tr>
<tr>
<td>dim25_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of a dimension in the Destination stage</td>
</tr>
</tbody>
</table>

**HPM_EXPD_POV**

The HPM_EXPD_POV view retrieves the states of each combination of dimension members included in a point of view (POV).

**Table 41  HPM_EXPD_POV**

<table>
<thead>
<tr>
<th>Field</th>
<th>SQL Server Data Type</th>
<th>Oracle Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>application_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the selected application</td>
</tr>
<tr>
<td>id</td>
<td>identity</td>
<td>integer(38,0)</td>
<td>Unique record ID</td>
</tr>
<tr>
<td>pov_dim1_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the first dimension member in the selected POV</td>
</tr>
<tr>
<td>pov_dim2_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the second dimension member in the selected POV</td>
</tr>
<tr>
<td>pov_dim3_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the third dimension member in the selected POV</td>
</tr>
<tr>
<td>pov_dim4_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the fourth dimension member in the selected POV</td>
</tr>
<tr>
<td>pov_state</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>The current status of the POV:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Draft</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Published</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Archived</td>
</tr>
</tbody>
</table>

**HPM_EXPD_DRIVER**

The HPM_EXPD_DRIVER view retrieves details about the driver, including driver type, display order and formula.

**Table 42  HPM_EXPD_DRIVER**

<table>
<thead>
<tr>
<th>Field</th>
<th>SQL Server Data Type</th>
<th>Oracle Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>application_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the selected application</td>
</tr>
<tr>
<td>id</td>
<td>identity</td>
<td>integer(38,0)</td>
<td>Unique record ID</td>
</tr>
<tr>
<td>Field</td>
<td>SQL Server Data Type</td>
<td>Oracle Data Type</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>----------------------</td>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the selected driver</td>
</tr>
<tr>
<td>description</td>
<td>nvarchar (255)</td>
<td>varchar2 (255 CHAR)</td>
<td>Description of the purpose for the selected driver</td>
</tr>
<tr>
<td>display_order</td>
<td>integer(38,0)</td>
<td>integer(38,0)</td>
<td>Display position of the driver within the list of all drivers in the model</td>
</tr>
<tr>
<td>formula</td>
<td>nvarchar (4000)</td>
<td>varchar2 (4000 CHAR)</td>
<td>Formula created for the driver by a user. This formula must be created using SQL syntax, and mathematically correct operations.</td>
</tr>
<tr>
<td>allow_idle_cost</td>
<td>nvarchar (1)</td>
<td>varchar2 (1 CHAR)</td>
<td>Flag determining whether idle costs are allowed for this driver:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• “Y” (Yes) specifying idle costs are allowed for this driver.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• “N” (No) specifying idle costs are not allowed for this driver.</td>
</tr>
<tr>
<td>priority</td>
<td>integer</td>
<td>number (38,0)</td>
<td>Enter the calculation priority of a driver so that allocations within a stage can be run in the specified order. The driver with the lowest priority is processed first. By default, the priority is set to 100, and the highest priority is 1. Numbers need not be sequential. Drivers with the same priority are executed in no particular order. Only whole, positive numbers are valid.</td>
</tr>
<tr>
<td>volume_formula</td>
<td>nvarchar (4000)</td>
<td>varchar2(4000 CHAR)</td>
<td>For Rate-Based drivers only, the formula which has both the Rate and Volume formulas defined.</td>
</tr>
<tr>
<td>dst_measure_member_name</td>
<td>nvarchar (2000)</td>
<td>varchar2(2000 CHAR)</td>
<td>Name of the measure on the Destination table</td>
</tr>
<tr>
<td>tdv_measure_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(2000 CHAR)</td>
<td>Measure that is used as the denominator in the allocation formula - Driver Value/TotalDriverValue (DV/TDV)</td>
</tr>
<tr>
<td>operation_type</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>The operation type selected for the driver:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Rate-based Allocation</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Ratio-based Allocation</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Calculated Measures</td>
</tr>
</tbody>
</table>

**HPM_EXPD_DRIVER_SEL**

The HPM_EXPD_DRIVER_SEL view retrieves details about the POV, and the driver selections.

**Table 43  HPM_EXPD_DRIVER_SEL**

<table>
<thead>
<tr>
<th>Field</th>
<th>SQL Server Data Type</th>
<th>Oracle Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>application_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the selected application</td>
</tr>
<tr>
<td>id</td>
<td>identity</td>
<td>integer(38,0)</td>
<td>Unique record ID</td>
</tr>
<tr>
<td>pov_dim1_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the first dimension member in the selected POV</td>
</tr>
</tbody>
</table>
### HPM_EXPD_DRIVER_EXCEP

The HPM_EXPD_DRIVER_EXCEP view retrieves details about the POV and source stage for the selected driver exceptions.

#### Table 44   HPM_EXPD_DRIVER_EXCEP

<table>
<thead>
<tr>
<th>Field</th>
<th>SQL Server Data Type</th>
<th>Oracle Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>application_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the selected application</td>
</tr>
<tr>
<td>id</td>
<td>identity</td>
<td>integer(38,0)</td>
<td>Unique record ID</td>
</tr>
<tr>
<td>pov_dim1_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the first dimension member in the selected POV</td>
</tr>
<tr>
<td>pov_dim2_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the second dimension member in the selected POV, if applicable</td>
</tr>
<tr>
<td>pov_dim3_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the third dimension member in the selected POV, if applicable</td>
</tr>
<tr>
<td>pov_dim4_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the fourth dimension member in the selected POV, if applicable</td>
</tr>
<tr>
<td>src_dim1_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the first dimension member in the source stage</td>
</tr>
<tr>
<td>src_dim2_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the second dimension member in the source stage, if applicable</td>
</tr>
<tr>
<td>src_dim3_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the third dimension member in the source stage, if applicable</td>
</tr>
<tr>
<td>src_dim4_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the fourth dimension member in the source stage, if applicable</td>
</tr>
<tr>
<td>src_dim5_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the fifth dimension member in the source stage, if applicable</td>
</tr>
<tr>
<td>driver_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the selected driver</td>
</tr>
</tbody>
</table>
**HPM_EXPD_ASGN_RUL_SEL**

The HPM_EXPD_ASGN_RUL_SEL view retrieves details about the assignment rule selections.

<table>
<thead>
<tr>
<th>Field</th>
<th>SQL Server Data Type</th>
<th>Oracle Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>application_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the selected application</td>
</tr>
<tr>
<td>id</td>
<td>identity</td>
<td>integer(38,0)</td>
<td>Unique record ID</td>
</tr>
<tr>
<td>pov_dim1_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the first dimension member in the selected POV</td>
</tr>
<tr>
<td>pov_dim2_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the second dimension member in the selected POV</td>
</tr>
<tr>
<td>pov_dim3_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the third dimension member in the selected POV</td>
</tr>
<tr>
<td>pov_dim4_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the fourth dimension member in the selected POV</td>
</tr>
<tr>
<td>src_dim1_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the first dimension member in the source stage</td>
</tr>
<tr>
<td>src_dim2_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the second dimension member in the source stage</td>
</tr>
<tr>
<td>src_dim3_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the third dimension member in the source stage</td>
</tr>
<tr>
<td>src_dim4_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the fourth dimension member in the source stage</td>
</tr>
<tr>
<td>src_dim5_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the fifth dimension member in the source stage</td>
</tr>
<tr>
<td>rule_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the rule controlling the selected assignment. The rule must be present in the target database.</td>
</tr>
</tbody>
</table>

**HPM_EXPD_CALCRULE_SNGLSRC**

The HPM_EXPD_CALCRULE_SNGLSRC view retrieves details about the single source calculation rule.

<table>
<thead>
<tr>
<th>Field</th>
<th>SQL Server Data Type</th>
<th>Oracle Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>application_name</td>
<td>VARCHAR2(80 CHAR)</td>
<td></td>
<td>Name of the selected application</td>
</tr>
<tr>
<td>id</td>
<td>identity</td>
<td>NUMBER(38,0)</td>
<td>Unique identifier for each row. This can come from any sequence generator.</td>
</tr>
<tr>
<td>name</td>
<td>nvarchar (80)</td>
<td>VARCHAR2(80 CHAR)</td>
<td>Name of the calculation rule.</td>
</tr>
<tr>
<td>description</td>
<td>nvarchar (255)</td>
<td>varchar2(255 CHAR)</td>
<td>Long description of the calculation rule.</td>
</tr>
<tr>
<td>pov_dim1_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the first dimension member in the selected POV</td>
</tr>
<tr>
<td>pov_dim2_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the second dimension member in the selected POV</td>
</tr>
<tr>
<td>Field</td>
<td>SQL Server Data Type</td>
<td>Oracle Data Type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>----------------------</td>
<td>---------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>pov_dim3_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the third dimension member in the selected POV</td>
</tr>
<tr>
<td>pov_dim4_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the fourth dimension member in the selected POV</td>
</tr>
<tr>
<td>calculation_sequence</td>
<td>integer</td>
<td>NUMBER(4,0)</td>
<td>Numeric value designating the relative order in which the calculation rule</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>will be processed during calculation. This must be &gt; 0 or you will get a</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>check constraint error during insert.</td>
</tr>
<tr>
<td>tag</td>
<td>nvarchar (255)</td>
<td>VARCHAR2(255)</td>
<td>Specify search tags for the calculation rule in this column.</td>
</tr>
<tr>
<td>enabled_flag</td>
<td>nvarchar (1)</td>
<td>VARCHAR2(1 CHAR)</td>
<td></td>
</tr>
<tr>
<td>first_sequence_priority</td>
<td>integer</td>
<td>NUMBER(38,0)</td>
<td>Designates the first driver sequence priority in the range of drivers to</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>be executed for this rule. This must be &gt; 0 or you will get a check</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>constraint error during insert.</td>
</tr>
<tr>
<td>last_sequence_priority</td>
<td>integer</td>
<td>NUMBER(38,0)</td>
<td>Designates the last driver sequence priority in the range of drivers to</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>be executed for this rule. This must be &gt;= FIRST_SEQUENCE_PRIORITY or you</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>will get a check constraint error during insert.</td>
</tr>
<tr>
<td>all_drivers_flag</td>
<td>nvarchar (1)</td>
<td>VARCHAR2(1 CHAR)</td>
<td>When Y ignore values in the First Sequence and Last Sequence priority</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>fields during calculation.</td>
</tr>
</tbody>
</table>

**HPM_EXPD_CALCRULE_CALCMSRS**

The HPM_EXPD_CALCRULE_CALCMSRS view retrieves details about the calculated measures rule.

**Table 47**  HPM_EXPD_CALCRULE_CALCMSRS

<table>
<thead>
<tr>
<th>Field</th>
<th>SQL Server Data Type</th>
<th>Oracle Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>application_name</td>
<td>nvarchar (80)</td>
<td>VARCHAR2(80 CHAR)</td>
<td>Name of the selected application</td>
</tr>
<tr>
<td>id</td>
<td>identity</td>
<td>NUMBER(38,0)</td>
<td>Unique identifier for each row. This can come from any sequence generator.</td>
</tr>
<tr>
<td>name</td>
<td>nvarchar (80)</td>
<td>VARCHAR2(80 CHAR)</td>
<td>Name of the calculation rule.</td>
</tr>
<tr>
<td>description</td>
<td>nvarchar (255)</td>
<td>varchar2(255 CHAR)</td>
<td>Long description of the calculation rule.</td>
</tr>
<tr>
<td>pov_dim1_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the first dimension member in the selected POV</td>
</tr>
<tr>
<td>pov_dim2_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the second dimension member in the selected POV</td>
</tr>
<tr>
<td>pov_dim3_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the third dimension member in the selected POV</td>
</tr>
<tr>
<td>Field</td>
<td>SQL Server Data Type</td>
<td>Oracle Data Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>----------------------</td>
<td>-----------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>pov_dim4_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the fourth dimension member in the selected POV</td>
</tr>
<tr>
<td>calculation_sequence</td>
<td>integer</td>
<td>NUMBER(4,0)</td>
<td>Numeric value designating the relative order in which the calculation rule will be processed during calculation. This must be &gt; 0 or you will get a check constraint error during insert.</td>
</tr>
<tr>
<td>tag</td>
<td>nvarchar (255)</td>
<td>VARCHAR2(255)</td>
<td>Specify search tags for the calculation rule in this column.</td>
</tr>
<tr>
<td>enabled_flag</td>
<td>nvarchar (1)</td>
<td>VARCHAR2(1 CHAR)</td>
<td></td>
</tr>
<tr>
<td>driver_name1</td>
<td>nvarchar (1)</td>
<td>VARCHAR2(1 CHAR)</td>
<td>Driver name to be associated with this calculation rule.</td>
</tr>
<tr>
<td>driver_name2</td>
<td>nvarchar (80)</td>
<td>VARCHAR2(80 CHAR)</td>
<td>Driver name to be associated with this calculation rule.</td>
</tr>
<tr>
<td>driver_name3</td>
<td>nvarchar (80)</td>
<td>VARCHAR2(80 CHAR)</td>
<td>Driver name to be associated with this calculation rule.</td>
</tr>
<tr>
<td>driver_name4</td>
<td>nvarchar (80)</td>
<td>VARCHAR2(80 CHAR)</td>
<td>Driver name to be associated with this calculation rule.</td>
</tr>
<tr>
<td>driver_name5</td>
<td>nvarchar (80)</td>
<td>VARCHAR2(80 CHAR)</td>
<td>Driver name to be associated with this calculation rule.</td>
</tr>
<tr>
<td>dest_assign_rule_name1</td>
<td>nvarchar (80)</td>
<td>VARCHAR2(80 CHAR)</td>
<td>Destination Assignment Rule to be associated with this calculation rule.</td>
</tr>
<tr>
<td>dest_assign_rule_name2</td>
<td>nvarchar (80)</td>
<td>VARCHAR2(80 CHAR)</td>
<td>Destination Assignment Rule to be associated with this calculation rule.</td>
</tr>
<tr>
<td>dest_assign_rule_name3</td>
<td>nvarchar (80)</td>
<td>VARCHAR2(80 CHAR)</td>
<td>Destination Assignment Rule to be associated with this calculation rule.</td>
</tr>
<tr>
<td>dest_assign_rule_name4</td>
<td>nvarchar (80)</td>
<td>VARCHAR2(80 CHAR)</td>
<td>Destination Assignment Rule to be associated with this calculation rule.</td>
</tr>
<tr>
<td>dest_assign_rule_name5</td>
<td>nvarchar (80)</td>
<td>VARCHAR2(80 CHAR)</td>
<td>Destination Assignment Rule to be associated with this calculation rule.</td>
</tr>
</tbody>
</table>

**HPM_EXPD_CALCRULE_MULTISRC**

The HPM_EXPD_MULTISRC view retrieves details about the multi source calculation rule.

**Table 48  HPM_EXPD_CALCRULE_MULTISRC**

<table>
<thead>
<tr>
<th>Field</th>
<th>SQL Server Data Type</th>
<th>Oracle Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>application_name</td>
<td>nvarchar(80)</td>
<td>VARCHAR2(80 CHAR)</td>
<td>Name of the selected application</td>
</tr>
<tr>
<td>id</td>
<td>identity</td>
<td>NUMBER(38,0)</td>
<td>Unique identifier for each row. This can come from any sequence generator.</td>
</tr>
<tr>
<td>name</td>
<td>nvarchar (80)</td>
<td>VARCHAR2(80 CHAR)</td>
<td>Name of the calculation rule.</td>
</tr>
<tr>
<td>description</td>
<td>nvarchar (255)</td>
<td>varchar2(255 CHAR)</td>
<td>Long description of the calculation rule.</td>
</tr>
<tr>
<td>Field</td>
<td>SQL Server Data Type</td>
<td>Oracle Data Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>----------------------</td>
<td>----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>pov_dim1_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the first dimension member in the selected POV</td>
</tr>
<tr>
<td>pov_dim2_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the second dimension member in the selected POV</td>
</tr>
<tr>
<td>pov_dim3_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the third dimension member in the selected POV</td>
</tr>
<tr>
<td>pov_dim4_member_name</td>
<td>nvarchar (80)</td>
<td>varchar2(80 CHAR)</td>
<td>Name of the fourth dimension member in the selected POV</td>
</tr>
<tr>
<td>calculation_sequence</td>
<td>integer</td>
<td>NUMBER(4,0)</td>
<td>Numeric value designating the relative order in which the calculation rule will be processed during calculation. This must be &gt; 0 or you will get a check constraint error during insert.</td>
</tr>
<tr>
<td>tag</td>
<td>nvarchar (255)</td>
<td>VARCHAR2(255)</td>
<td>Specify search tags for the calculation rule in this column.</td>
</tr>
<tr>
<td>enabled_flag</td>
<td>nvarchar (1)</td>
<td>VARCHAR2(1 CHAR)</td>
<td>Driver name to be associated with this calculation rule.</td>
</tr>
<tr>
<td>driver_name</td>
<td>nvarchar (80)</td>
<td>VARCHAR2(80 CHAR)</td>
<td>Driver name to be associated with this calculation rule.</td>
</tr>
<tr>
<td>src_assign_rule_name1</td>
<td>nvarchar (80)</td>
<td>VARCHAR2(80 CHAR)</td>
<td>Source assignment rule to be associated with this calculation rule.</td>
</tr>
<tr>
<td>src_assign_rule_name2</td>
<td>nvarchar (80)</td>
<td>VARCHAR2(80 CHAR)</td>
<td>Source assignment rule to be associated with this calculation rule.</td>
</tr>
<tr>
<td>src_assign_rule_name3</td>
<td>nvarchar (80)</td>
<td>VARCHAR2(80 CHAR)</td>
<td>Source assignment rule to be associated with this calculation rule.</td>
</tr>
<tr>
<td>src_assign_rule_name4</td>
<td>nvarchar (80)</td>
<td>VARCHAR2(80 CHAR)</td>
<td>Source assignment rule to be associated with this calculation rule.</td>
</tr>
<tr>
<td>src_assign_rule_name5</td>
<td>nvarchar (80)</td>
<td>VARCHAR2(80 CHAR)</td>
<td>Source assignment rule to be associated with this calculation rule.</td>
</tr>
<tr>
<td>dest_assign_rule_name1</td>
<td>nvarchar (80)</td>
<td>VARCHAR2(80 CHAR)</td>
<td>Destination assignment rule to be associated with this calculation rule.</td>
</tr>
<tr>
<td>dest_assign_rule_name2</td>
<td>nvarchar (80)</td>
<td>VARCHAR2(80 CHAR)</td>
<td>Destination Assignment Rule to be associated with this calculation rule.</td>
</tr>
<tr>
<td>dest_assign_rule_name3</td>
<td>nvarchar (80)</td>
<td>VARCHAR2(80 CHAR)</td>
<td>Destination assignment rule to be associated with this calculation rule.</td>
</tr>
<tr>
<td>dest_assign_rule_name4</td>
<td>nvarchar (80)</td>
<td>VARCHAR2(80 CHAR)</td>
<td>Destination assignment rule to be associated with this calculation rule.</td>
</tr>
<tr>
<td>dest_assign_rule_name5</td>
<td>nvarchar (80)</td>
<td>VARCHAR2(80 CHAR)</td>
<td>Destination assignment rule to be associated with this calculation rule.</td>
</tr>
</tbody>
</table>
Generating Statistics for a Detailed Profitability Application

The Detailed Application Model Calculation Statistics script can be used with Detailed Profitability applications to generate model and model calculation statistics that can be useful in understanding and debugging your model.

The script creates a table within the HPCM Product Schema and stores statistics useful for troubleshooting model calculated results problems, for describing the shape and characteristics of the model, and for evaluating model calculation performance.

The dpmodelcalcstats.sql script is located within

`%EPM_ORACLE_HOME%\OPatch\HPCM_11_1_2_2_351_14845052\files\products\Profitability\database\Common\Oracle`

after the 11.1.2.2.351 patch has been installed. It is currently only available for Oracle database installations.

The script is designed to be open and to run within SQL Developer.

To generate statistics for the requested application:

1. Open `dpmodelcalcstats.sql` within Oracle SQL Developer.
3. Move the cursor to the first line of the script.
4. Click on Run Script or press the F5 key.
5. Enter the name of the application where model calculation statistics are requested.

The script creates a table within the product schema using the following naming convention:

`HPMD_<Application Name>_DIAGSTAT_<Unique Number>`

A short description of the columns within the HPMD_BksDP20_DIAGSTAT_4129 table:

- **APPLICATION**: The name of the application.
- **ITEM**: A short description of the content within the row.
- **RATIO_DRV_TOT**: The total number of items related to all of the Ratio-based drivers.
- **RATE_DRV_TOT**: The total number of items related to all of the Rate-based drivers.
- **OBJC_DRV_TOT**: The total number of items related to all of the Object Calculation drivers.
- **TOTAL**: The total number of items related to all of the assignment.
- **NAME**: The name of the artifact or object related to the item.

These types of diagnostic statistic items are included within the table:

**Allocation Table**: The name of the allocation table with all of the assignment details for the specified application. If the count is greater than 1, then results are likely incorrect because clear calculations has not been run.
**Number of Assignments:** The number of assignments identified within the model for the specified application.

**No of source Nodes:** The number of source nodes (intersections) identified within the model.

**Number of Assignments with Data:** The number of assignments where one or more rows within the source stage business object match a potential source node.

**Number of row updates:** The total number of rows altered by the update statements performed for all of the assignments.

**Number of Assignment Rules:** The number of assignment rule artifacts defined within the model with at least one assignment referencing the assignment rule.

**Number of Drivers:** The number of driver artifacts defined within the model with at least one assignment referencing the driver.

**Maximum Destination Row Count:** The maximum number of rows updated within the destination by an assignment.

**Median Destination Row Count:** The median number of rows updated within the destination by an assignment.

**Deviation of Destination Row Count:** The standard deviation for rows updated within the destination by an assignment.

**Driver Count:** There is one driver count item per driver referenced by at least one assignment. The item includes name of the driver artifact and the number of times referenced by an assignment.

**Assignment Rule Count:** There is one assignment rule count item per assignment rule referenced by at least one assignment. The item includes name of the assignment rule artifact and the number of times referenced by an assignment.
Part IV

Working with Management Ledger Applications

In Working with Management Ledger Applications:

- Understanding Management Ledger Profitability Application Architecture
- Understanding Management Ledger Dimensions
- Performing Other Application Management Tasks
In This Chapter

About the Management Ledger Architecture .......................................................... 175

About the Management Ledger Architecture

Accessed through EPM Workspace, Profitability and Cost Management is an analytical tool that resides on top of Essbase. Profitability and Cost Management 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 3 on page 175). For a description of Management Ledger Profitability, see “Management Ledger Profitability” on page 19.

Figure 3  Management Ledger Profitability Product Architecture

The model metadata from Workspace and EPM Architect is used in the creation of the Profitability and Cost Management model, and the calculated results can be output in a variety of reporting and analysis tools.
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 an aggregate storage (ASO) database for both calculation and reporting.

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

Results may be viewed in reporting and analysis tools, such as Oracle Hyperion Financial Reporting, Oracle Hyperion Web Analysis, and Oracle Smart View for Office.
About Management Ledger Dimensions

Profitability and Cost Management uses dimensions and members created in Performance Management Architect or Essbase and the Profitability Applications feature to represent many of the structural elements of the business model.

A dimension type is a dimension property that enables the use of predefined functionality. The specific characteristics of the dimension type manage the behavior and functions of the dimension. Because Profitability and Cost Management, Performance Management Architect and other EPM Workspace products may share certain dimension types, you can leverage the functionality of dimensions for different products.

Management Ledger Profitability applications have the following dimensions:

- “Management Ledger System Dimensions” on page 178
- “Business Dimensions” on page 180
- “POV Dimensions” on page 181
- “Attribute Dimensions” on page 181
- “Alias Dimensions” on page 182

Management Ledger Dimension Requirements

The database outline provides the data structure for the model, and includes calculation instructions and formulas. Dimensions in the Essbase outline are hierarchical. Data is stored at dimension intersections. The following are Management Ledger Profitability dimension requirements:
Applications or models must contain at least one POV dimension and can have up to four POV dimensions.

Applications must contain one and only one system dimension named **Rule**.

Users can edit and add more members to the Rule dimension. For example, R1001 through R1500. Optionally, they can also trim that dimension. The Calculation Programs member in the Rule dimension is not editable.

Applications must contain one and only one system dimension named **Balance**.

System dimension members in the Balance dimension cannot be edited. Users can, however, add alternate hierarchies.

There should be at least one business dimension with no duplicate members in the primary hierarchy of business dimensions.

**Caution!** Members must not be repeated within the same dimension. However, members can be repeated across several dimensions.

Figure 4 on page 178 shows a sample Essbase outline of a Management Ledger Profitability database, shown on the Essbase console.

---

**Management Ledger System Dimensions**

**Subtopics**

- Management Ledger Rule Dimension
- Management Ledger Balance Dimension

Management Ledger applications must contain two system dimensions: Rule and Balance. These system dimensions are populated from Performance Management Architect or Essbase into
Profitability and Cost Management. For additional information about the Rule and Balance dimensions, see the listed sections and the *Oracle Hyperion Profitability and Cost Management Administrator’s Guide*.

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

## Management Ledger Rule Dimension

The Rule dimension contains definitions of calculation rules for Management Ledger applications. *Figure 5 on page 179* shows the outline of the Rule dimension in the Essbase console. This is an illustration of the Calculation Rules members.

*Figure 5  The Management Ledger Rule Dimension Outline*

Users can delete and add rules to Calculation Rules, but the only rule they can apply is NoRule. All the others are reserved for system use.

*Figure 4 on page 178* shows an additional member, Calculation Programs. This member is controlled by the system and is not editable by users.

## Management Ledger Balance Dimension

*Figure 6 on page 180* shows the outline of the Balance dimension in the Essbase console.
Users can add data to the Input member of Net Balance. The remainder of the members reflect inputs and outputs determined by rule sets and rules. Adjustments are the result of driver calculations, allocations are the result of rule allocations, and offsets result from rule offset definitions. See the Oracle Hyperion Profitability and Cost Management User’s Guide for information about rule sets, rules, and their definitions.

Data held in intersections with these members is visible in the Rule Balancing screen.

### Business Dimensions

Business dimensions describe the business-specific objects within 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 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 reuse the dimension member and hierarchies that were defined for other applications, such as Oracle Hyperion Planning.
Note: This dimension type does not apply to aggregate storage outlines.

When creating a business dimension for a Management Ledger application, the following requirements apply:

- 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 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 Dimension HierarchyType to “Enabled”, the first Gen2 member HierarchyType to “Stored” and the Gen2 member with alternative hierarchy with shared members to “Dynamic”.

These alternate hierarchies are not visible in Profitability and Cost Management modeling screens, and can only be viewed in Essbase.

POV Dimensions

POV dimensions indicate a specific point of view or version of the model, such as year, scenario, or period. The dimension can be customized to reflect the requirements of your organization. For example, the POV may consist of quarters, months, seasonal groupings, and so on.

At least one POV dimension is required for each model, but you can create up to four POV dimensions. The POV dimensions are set in the Performance Management Architect Property grid for the model.

A Version dimension is also available, and is used to create another instance of your model. This version can be modified to enable you to experiment with strategies or business options to play “what-if” scenarios. By modifying the version, you can implement features, and compare results to determine the best course of action.

Attribute Dimensions

Attribute dimensions are a special type of dimension that are associated with a business dimension, and contain members that can be used to classify members of another, associated dimension. Attribute dimensions describe characteristics of data, such as the size and 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 assignment rules are created.

Note: For naming restrictions, see Appendix F, “Essbase Naming Conventions”.
Two Attribute dimension types exist in Performance Management Architect:

- **Attribute Dimensions:**
  - The attribute can be created using different structures, such as Boolean, Date, Numeric, and Text.
  - An attribute has a hierarchy, and the hierarchies can be rolled up for aggregate values.
  - Only one attribute from a given attribute dimension can be associated with one member.

- **User-Defined Attribute Dimensions (UDAs):**
  - The attribute can only be created using Text.
  - A UDA does not have a hierarchy and cannot be easily used in reports to generate sums.
  - Multiple UDAs can be associated with one member.

Each type of attribute dimension offers different advantages, depending on your model and reporting needs. For detailed information about working with attribute dimensions, see the *Oracle Essbase Database Administrator’s Guide*.

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.

For naming restrictions, see Appendix F, “Essbase Naming Conventions”.

### Alias Dimensions

Aliases are alternate names, descriptions, languages, or other items that help to define dimensions. For example, you may refer to a customer number in the system, but you can assign an alias that displays the company name on the screen, to make it easier to identify that client. You can assign one or more aliases to accounts, currencies, entities, scenarios, periods, versions, years, and user-defined dimension members.

**Note:** Duplicate member names or aliases are not allowed within the same dimension.

For Profitability and Cost Management, the alias must be set in Performance Management Architect. For detailed instructions on creating Alias dimensions, see the *Oracle Hyperion Enterprise Performance Management Architect Administration Guide*.

**Caution!** If an Alias association is deleted in Performance Management Architect, it is not deleted from the model until the application is redeployed into Profitability and Cost Management.

When installation is complete, a “Default” alias table is available. After redeployment, you can view the alias on all screens that use the Management Ledger member selector, such as the Rule Definition screen.
Management Ledger 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 a 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 56.

The Dimension Sort Order settings for the model are validated in Performance Management Architect. See “Modifying the Standard Profitability Application Settings” on page 73.

You can also set the density for dimensions from the Dimension Sort Order Settings dialog box.

Dimension Sort Order Recommendations

Oracle recommends that you set the dimension sort using the following recommendations:

- 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, unique, and greater than or equal to 1.
- Rule dimension is set to 1, by default.
- Balance dimension is set to 2, by default.
- 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 Management Ledger 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 Recommendations” on page 183.

There are two ways to set the dimension sort order in Performance Management Architect:

- Using the Dimension Performance Settings dialog box to set the sort order for all dimensions at one time
- Using the Property Grid in the Dimension Library to set individual dimension sort order for one dimension at a time
To set the Dimension Sort Order using the Dimension Performance Settings dialog box:

1. From EPM Workspace, select Navigate, then Administer, and then Dimension Library to display the Shared Library.

2. Right-click the name of the application, and select Dimension Performance Settings.

   **Note:** The following figure shows Standard Profitability dimensions.

3. Select a dimension, and use the Up 🔄 and Down 🔄 arrows to move each dimension into the correct sort order, as described in “Setting the Dimension Sort Order” on page 56. The number of the original position for the dimension is displayed under Sort Order.

4. **Optional:** Under Density for the selected dimension, double-click the cell to display the options, and then select the appropriate density for the dimension.

5. Click OK.

To set Dimension Sort Order for individual dimensions:

1. From EPM Workspace, select Navigate, then Administer, and then Dimension Library.

2. In the Profitability and Cost Management application, select the dimension for which you want to set the Dimension Sort Order.

3. In the Property Grid, select the Dimension Sort Order property, and under Value, enter the required number for the sort order. See “Setting the Management Ledger Dimension Sort Order” on page 183.

4. Click the Save icon 📄.
Performing Other Application Management Tasks

In This Chapter

Managing Management Ledger Profitability Applications with Performance Management Architect .......................................................... 185
Validating and Deploying Management Ledger Applications Using the Wizard ......................... 191

Managing Management Ledger Profitability Applications with Performance Management Architect

Working with Management Ledger Applications and Dimensions

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

Note:

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, 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:

- “Optimizing Dimension Settings for Essbase” on page 60
- “Importing Metadata” on page 61
- “Profitability and Cost Management Dimension and Member Properties” on page 62
- “Creating Standard Profitability Applications” on page 68
- “Validating and Deploying the Standard Profitability Application in Performance Management Architect” on page 77
- “Synchronizing Data” on page 82

**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.

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 Enterprise Performance Management System User Security Administration Guide.

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 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.

From the Dimension Library, you can edit the properties of these:

- 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 Administration 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 12, “Standard Profitability and Cost Management Dimension and Member Properties”, which displays the following information:

- The Property Label, which provides a more readable display name for the property. If applicable, the associated database type is appended to the name (ASO or BSO). If no database 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 Administration 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.

Setting Hierarchy Type Property

The ‘Hierarchy Type’ property applies only to Aggregate Storage Databases in Essbase.

To be able to use other consolidation type symbols (other than ADDITION and IGNORE) the hierarchy type should be set to ‘DYNAMIC.’

Note: If a dimension member has a formula, the hierarchy type must be set to ‘DYNAMIC.’

To set the Hierarchy Type:

1. From EPM Workspace, select Navigate, then Administer, and then Dimension Library.
2. Under the application, select some dimension (except Alias, Attribute, UDA).
3. In the Property Grid, select the appropriate Hierarchy Type property:
   - Select Stored for any dimension whose dimension members use the following consolidation symbols:
Creating Management Ledger Profitability Applications

You can create Management Ledger 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.

To create Management Ledger Profitability applications:

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

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

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 special characters, including “&” (ampersands). See Appendix F, “Essbase Naming Conventions”

5. Under Type, select Profitability.
6 Optional: 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 Standard Profitability Dimensions Manually” on page 72.

8 Optional: To automatically create all required dimensions, select Auto Create Local Dimensions.

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 Under Create As, select the type of Profitability and Cost Management application to create, in this case Management Ledger.

10 Click Next. See “Adding Management Ledger Profitability Dimensions Using the Wizard” on page 189.

**Adding Management Ledger Profitability 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:

- Balance
- Rule
- POV Dimension
- Business Dimension
- Alias Dimension (Optional)
- Attribute Dimension (Optional)
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).

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 view restricted words and characters for names, see Appendix F, “Essbase Naming Conventions”.

➢ To define dimensions:

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

![Figure 7 The Management Ledger Dimension Selection Tab, Automatically Populated]

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].
c. In the **Add New Dimension** dialog box, 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 required 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.

7 **Click Next.**

The Application Settings screen is displayed.

8 **Modify the application settings, as required.** See “**Modifying the Standard Profitability Application Settings**” on page 73 and adjust procedures for Management Ledger Profitability applications.

**Note:** If you want to add dimensions manually, see “**Adding Standard Profitability Dimensions Manually**” on page 72 and adjust the process for Management Ledger dimensions.

---

**Validating and Deploying Management Ledger Applications Using the Wizard**

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

➢ **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.**
To validate using Performance Management Architect, see “Validating and Deploying Applications through Performance Management Architect” on page 80.
In Appendices:

- Creating the First Profitability and Cost Management Application with Performance Management Architect
- Creating Applications Using the Profitability Applications Console
- Importing Data into Profitability and Cost Management
- Migrating Data Using EPM System Lifecycle Management
- Backing Up Profitability and Cost Management Components
- Essbase Naming Conventions
Creating the First Profitability and Cost Management Application with Performance Management Architect

In This Appendix

About Creating the First Application with Performance Management Architect .................................. 195
Steps to Create the First Application with Performance Management Architect .......................... 195

About Creating the First Application with Performance Management Architect

Profitability and Cost Management is an integral part of EPM Workspace, and uses common software to manage the application and security. After installation, an administrator or user with appropriate security provisioning must perform a number of steps to create the first Profitability and Cost Management application. After the application is created, data must be imported into Profitability and Cost Management.

Note: The instructions in this section describe how to create the first Profitability and Cost Management application using Performance Management Architect.

If you are not able to use Performance Management Architect, you can create dimensions in Essbase or a flat file (Management Ledger applications only), and then use Profitability Applications Console within EPM Workspace to complete the application. For instructions, see Appendix B, “Creating Applications Using the Profitability Applications Console.”

Steps to Create the First Application with Performance Management Architect

1. Complete the installation of the following software components:
   - EPM Workspace
   - Shared Services
Note: This list represents the minimum installation required to use Profitability and Cost Management. However, you may install additional products at any time.

For complete installation instructions, see:

- *Oracle Enterprise Performance Management System Installation Start Here*
- *Oracle Enterprise Performance Management System Installation and Configuration Guide*

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

For detailed instructions on creating dimensions, see the *Oracle Hyperion Enterprise Performance Management Architect Administration Guide*. For information on required dimensions, see relevant sections for each type of Profitability application in this Guide.

Note: Although multibyte characters are allowed in Performance Management Architect, they are not supported in Profitability and Cost Management.

For non-Unicode Standard Profitability applications, do not use multibyte characters for the Calculation or Reporting database names. Use ASCII characters only for the database names.

If the sum of the lengths of the prefix and the dimension name exceeds 80 characters, deployment to Essbase fails.

3 Add the dimensions, and then select **Navigate**, then **Administer**, and then **Application Library**.

4 From the Application Library, select **File**, then **New**, and then **Application** to create a new Standard, Detailed, or Management Ledger Profitability and Cost Management application. Select dimensions for Profitability and Cost Management from the Dimension Library.

Note: In instances where the name of a new Profitability and Cost Management application contains foreign characters, which may result in longer names, you may not be able deploy the application in Performance Management Architect. In that case, create the application using a shorter version of the application name, or enter the name in English.

For additional instructions on creating a Profitability and Cost Management application, see the *Oracle Hyperion Enterprise Performance Management Architect Administration Guide*.

5 From the Application Library, right-click the new application name, and select **Validate**.
The Validation task is submitted to verify the metadata. Click the link on the job status message to view the Job Console for the validation task, and view any warnings or errors in the log file under Attachments. You must correct any errors before deploying the application.

6 From the Application Library, right-click the new application name, select Deploy, and then perform the following steps:
   
a. From Deploy, under Shared Services Project, select the name of the Profitability and Cost Management application group.

   **Note:** The Profitability and Cost Management application group is not displayed the first time you perform this action. You must create the Profitability and Cost Management application group or project in Shared Services. See the Oracle Enterprise Performance Management System User Security Administration Guide.

   b. Click Deploy.

   c. From the Job Task window, click the link to for the Job ID to display the job status.

   d. Under Detail on the Job Task window, when the completion message is displayed, the new application has been created and deployed to Profitability and Cost Management and is available for selection.

   See the Oracle Hyperion Enterprise Performance Management Architect Administration Guide.

7 On the EPM Workspace main menu, select Navigate, then select Applications, then Profitability. Click Refresh and then select the new application.

8 From Task Areas, select Manage Models, and then Import Staging Tables to import the data that you require to begin building a Standard or Detailed Profitability model, as outlined earlier in this Guide. Staging tables are not used with Management Ledger Profitability applications.
Creating Applications Using the Profitability Applications Console

In This Appendix

Guidelines for Using the Profitability Applications Console ................................................. 199
Displaying the Profitability Applications Console ................................................................. 200
Creating Profitability and Cost Management Applications in the Profitability Applications Console ............................................................................................................. 201
Working with Applications in the Profitability Applications Console ................................. 204
Preparing Templates and Flat Files for Creating and Updating Management Ledger Profitability Applications .............................................................. 210
Creating an Essbase Master Cube for Profitability and Cost Management .................... 214

Guidelines for Using the Profitability Applications Console


The Profitability Applications Console provides the following alternative ways to create and manage Profitability and Cost Management applications and dimensions without using Performance Management Architect:

- Creating a Profitability and Cost Management application using the Actions, New option with Dimension Source set to Master Cube. This type of application uses Essbase as the source of its dimensions (“Creating Applications with Dimensions from an Essbase Master Cube” on page 201).

- Creating an application using the Actions, New option with Dimension Source set to File. This type of application uses specially formatted flat files containing dimension definitions (for Management Ledger applications only; see “Creating Management Ledger Applications with Dimensions from Flat Files” on page 203).

- Using Actions, Import Template to create applications of File type with template files containing dimensions, metadata, and other artifacts created by exporting templates from existing Management Ledger applications (for Management Ledger applications only; see “Creating Management Ledger Applications by Importing Template Files” on page 203).
Note: Import and export of template files for Management Ledger applications is introduced in release 11.1.2.4.110 and is not available in earlier releases.

Whether you use Performance Management Architect or the Profitability Applications Console to create Profitability and Cost Management applications, there is no difference in how the applications work. However, you can only modify application dimensions by using the original dimension creation system (Essbase for Deploy Origin type Master Cube, or flat files for Deploy Origin type File).

To open and view the Profitability Applications Console, see “Displaying the Profitability Applications Console” on page 200.

Displaying the Profitability Applications Console

➢ To open and view the Profitability Applications Console:

1 From EPM Workspace, select Navigate, then Administer, and then Profitability Applications.

The Profitability Applications Console is displayed, showing the Applications ( kho) and Job Library ( kho) tabs (Figure 8 on page 200). The Profitability Applications Console lists any existing applications, their type, whether they were deployed from EPM Architect, a Master Cube, or a File, and whether they are enabled.

Tip: Click kho to view summary information for an existing application. Click kho to view dimensions included in the selected application.

Figure 8 Profitability Applications Console, Applications Tab

2 Select Actions to perform a variety of operations on the selected application or create a new application.
Creating Profitability and Cost Management Applications in the Profitability Applications Console

Subtopics

- Creating Applications with Dimensions from an Essbase Master Cube
- Creating Management Ledger Applications with Dimensions from Flat Files
- Creating Management Ledger Applications by Importing Template Files

“Guidelines for Using the Profitability Applications Console” on page 199 describes several ways to create applications using the Profitability Applications Console. The listed topics describe how to create an application using these alternatives. The following instructions assume that you have already defined dimensions as explained in “Creating an Essbase Master Cube for Profitability and Cost Management” on page 214 or “Preparing Templates and Flat Files for Creating and Updating Management Ledger Profitability Applications” on page 210.

For additional actions you can perform in the Profitability Applications Console, see “Working with Applications in the Profitability Applications Console” on page 204.

Creating Applications with Dimensions from an Essbase Master Cube

To create applications in the Profitability Applications Console using dimensions from an Essbase Master Cube:

1. From EPM Workspace, select Navigate, then Administer, and then Profitability Applications (Figure 8 on page 200).

2. Select Actions, and then New to create a new application (Figure 9 on page 202).
3 In the New Application screen, enter the following information and then select Next:
   - **Application Name** for the application
   - Optional: **Description** of the application
   - Select the **Instance Name** from the drop-down list
   - The address of the **Web Server** is displayed
   - The **Essbase Application Server** for the application
   - The **Shared Services Project** for the application
   - The application type; **Management Ledger** is the default but **Standard Profitability** and **Detail Profitability** are also available
   - The **Dimension Source**: **Master Cube** is the default

4 Select **Master Cube** as the **Dimension Source**, select the name of the Essbase Master Cube to provide dimensions, and then select dimensions to include in the application.

5 Click **Finish**.

When you click **Finish** with **Master Cube** as the **Dimension Source**, the following actions are initiated:
   - A new task flow begins to create the new application. Results can be viewed on the Job Library tab.
   - Dimensions in the source Essbase database are validated. If there are validation issues, the task flow ends and an error message appears on the Job Library tab. Click the error link for details. Fix any validation errors, and repeat the steps to create the new application.

**Tip:** As you work, click **Refresh** on the Job Library tab and then the Applications tab when the job is successful.
After the validation is complete, the new application is added to the Applications tab with Deploy Origin of Master Cube.

Creating Management Ledger Applications with Dimensions from Flat Files

To create a Management Ledger application in the Profitability Applications Console using dimensions from flat files:

1. From EPM Workspace, select Navigate, then Administer, and then Profitability Applications (Figure 8 on page 200).
2. Select Actions, and then New to create a new application (Figure 9 on page 202).
3. In the New Application screen, enter the following information and then select Next:
   - Application Name for the application
   - Optional: Description of the application
   - Select the Instance Name from the drop-down list
   - The address of the Web Server is displayed
   - The Essbase Application Server for the application
   - The Shared Services Project for the application
   - The application type; Management Ledger is the default but Standard Profitability and Detail Profitability are also available
   - The Dimension Source, Master Cube is the default and File is also available for Management Ledger applications
5. Enter names to use for the Rule and Balance dimensions in the new Management Ledger application. Then, select Unicode if used. Click Finish. The application is added to the Applications tab with Deploy Origin of File.
6. Perform the steps in “Updating Application Dimensions” on page 206 to add additional dimensions to the application:
7. When dimensions are complete, use the Validate and Enable option so rules can be added (“Performing Other Application Actions” on page 205).

Creating Management Ledger Applications by Importing Template Files

“Exporting Templates” on page 210 describes how to export Management Ledger applications to template files for application migration and backup purposes. This section describes how to import those files to create new applications with application metadata, dimensions metadata, and model artifacts previously exported from another application.
To create a Management Ledger application with a template file:

1 Export a template file as described in “Exporting Templates” on page 210.

2 On the Applications tab of the Profitability Applications Console, select Actions, and then Import Template.

3 Select a location for the template file, either on a server or your local computer.

   Note: Files loaded from the server must first be copied to the import_export folder. Note that this folder is the same one used for LCM (lifecycle management), such as `<MIDDLEWARE_HOME/user_projects/epmsystem1/import_export`.

4 Browse to select a file with .zip extension, and then click Next.

   If the file is a valid format, the import begins. Otherwise, an error message is displayed.

   You can click the Job Library tab, and then Refresh to check import status.

Working with Applications in the Profitability Applications Console

Subtopics

- Editing Application Descriptions and Shared Services Projects
- Performing Other Application Actions
- Updating Application Dimensions
- Viewing Tasks in the Profitability Applications Console Job Library

The Applications tab of the Profitability Applications Console lists Profitability applications created in Performance Management Architect along with those created from flat files and template files (Deploy Origin of type File), and Essbase Master Cubes.

Note: The applications created in Performance Management Architect only can be reregistered. Other update tasks must be performed in Performance Management Architect.

You can edit the selected application’s Description and Shared Services Project (“Editing Application Descriptions and Shared Services Projects” on page 205).

For applications created in the Profitability Applications Console, you can perform the following tasks using the Actions menu: Delete, Duplicate, Validate and Enable, Re-Register, Metadata Validation, and Update Dimensions. For Management Ledger applications, you can also Import Template and Export Template. For more information, see the topics listed at the beginning of this section.
Editing Application Descriptions and Shared Services Projects

Only the application’s Description and Oracle Hyperion Shared Services Project can be modified in the Overview window. If you want to change any other information related to the application that is displayed in the Overview window, you must create a new application.

To edit applications:

1. From EPM Workspace, select Navigate, then Administer, and then Profitability Applications.
   
   The Profitability Applications Console opens, showing the Applications and Job Library tabs.

2. Select the application to be modified, and then click the Overview button.

3. View and modify information as required:
   
   - Modify the Description, if required
   
   - Select a different Shared Services Project from the drop-down list.

4. Click Save.

To reregister an application, see “Performing Other Application Actions” on page 205.

Performing Other Application Actions

For applications created in the Profitability Applications Console, you can perform the following tasks from the Actions menu: Delete, Duplicate, Validate and Enable, Re-Register, Metadata Validation, and Update Dimensions.

Note: For Management Ledger applications, you can also export and import template files (“Exporting Templates” on page 210 and “Creating Management Ledger Applications by Importing Template Files” on page 203)

To perform an application action in the Profitability Applications Console:

1. From EPM Workspace, select Navigate, then Administer, and then Profitability Applications.

2. On the Applications tab, select the target application.

3. Select Actions, and then one of the following options:
   
   - Delete — Deletes the selected application
     
     Ensure no other users require this application before deleting it.

   - Duplicate — Copies the selected application
     
     You will be asked to provide a name for the new application.

   - Validate and Enable — Validates the selected application and enables it when valid
Results are displayed on the Job Library tab . If a validation error occurred, click the link for details. You can create a file of errors for easier correction.

- **Re-Register** — Refreshes the shared services registration for the selected application
  This option is available for applications created in Performance Management Architect as well as Profitability Applications Console.

- **Metadata Validation** — Runs cross-dimension validations on the deployed data for the selected application but does not enable valid applications
  Select **Validate and Enable** to enable valid applications.

- **Update Dimensions** — Add or remove dimension members for the selected application
  For details, see “Updating Application Dimensions” on page 206.

4. **Respond to any confirmation prompts.**
The selected action is performed.

**Tip:** View and refresh the Job Library tab to check the current status of a Profitability Applications Console job (“Viewing Tasks in the Profitability Applications Console Job Library” on page 207).

---

### Updating Application Dimensions

If any changes to dimensions or dimension members are required in an application with Deploy Origin of **Master Cube** – such as adding, deleting, changing, or renaming – you must update the dimensions in the Essbase Master Cube and then update the Profitability and Cost Management application.

**Caution!** If you need to rename or delete dimensions in an Essbase Master Cube, consider creating a separate Essbase Master Cube so that your existing Profitability and Cost Management applications can still be updated if necessary. Also, you cannot update dimensions using the Profitability Applications Console if the name of the Essbase Master Cube database is not the same as the name of the Profitability and Cost Management application that is based on it.

Whenever the Profitability Applications Console is used to update dimensions, a Job entry is automatically created in the Job Library to validate and update the application.

To update Essbase dimensions in a Profitability application created in the Profitability Applications Console:

1. **From EPM Workspace, select Navigate, then Administer, and then Profitability Applications.**
2. **In the Applications tab, select the application that contains the dimensions and dimension members to update.**
3. **Select Actions, and then Update Dimensions.**
Typically, you would perform a Pre-Update Analysis and then select Update Dimensions.

4 **Optional:** Under Pre Update Analysis, select Validate Dimensions to verify that each of the selected changes are valid. This option does not update the dimensions. If all dimensions have been selected, this option also runs the application level dimensions validations.

Validation errors and the results of the impact analysis are displayed in the Job Library.

5 **Optional:** Under Pre Update Analysis, select Impact Analysis to perform a comparison between the model being deployed and the existing model in Profitability and Cost Management. If Impact Analysis is selected, the Validate Dimensions option is selected automatically.

The following information is displayed:

- New Members
- Deleted Members
- Re-Parented Members
- Members with Level0 change
- Impacts on referencing model artifacts

Validation errors and the results of the impact analysis are displayed in the Job Library.

6 **Optional:** Check Update Dimensions to update the selected dimensions with the changes.

7 Under Dimensions, do one of the following:

- If you are updating an application with Deploy Origin of Master Cube, select the dimensions to update.

  **Note:** Dimension members that are not selected will be omitted or deleted if they were previously included.

- If you are using flat files to add or update dimensions to a Management Ledger application with Deploy Origin of File, browse to select a flat file for the dimension to update.

8 Click OK. Any exceptions are reported in the Job Library.

If you selected Pre Update Analysis, the modified dimensions are validated but not updated.

If you selected Update Dimensions, the selected dimension members are added, updated, or deleted according to dimension selections, or the selected flat file is imported.

**Note:** If you are updating with flat files, repeat necessary steps until all flat files have been imported. Be sure to use the Validate and Enable option to ensure that files were imported correctly.

**Viewing Tasks in the Profitability Applications Console Job Library**

The Job Library tab provides a list of all jobs created in the Profitability Applications Console.
To view the Job Library:

1. From EPM Workspace, select **Navigate**, then **Administer**, and then **Profitability Applications**.

2. Select the **Job Library** tab ( ).

Figure 10  The Profitability Applications Console Job Library Screen

The **Job Library** screen contains the following controls:

- **Export to Excel** button — Saves the Job Library table to a Microsoft Excel file
- **Stop** button — Cancels the selected job
- **Delete** button — Removes the selected job from the Job Library table
- **Refresh** button — Updates the Job Library table with the most recent job information
- **View** menu — Hide, show, and rearrange the order of columns; detach and reattach the Job Library table; display and hide the Query by Example boxes at the top of each column
- **Filter** button — , displays and hides Query by Example boxes at the top of each column
  
Enter text to match into a Query by Example box to select specific entries in the Job Library table.
- **Detach** button — , floats the Job Library table in its own window; click to attach it again

3. View the information for each job:

- **Job Id** is a sequential identification number assigned by the system.
- **User** identifies the user name of the individual who submitted the task for processing.
- **Application Name** displays the name of the application for which the task is being run.
- **Start Date/Time** displays the date and time on which the job was submitted or is scheduled to be run.
- **End Date/Time** displays the date and time when the job stopped, successful or not.
- **Elapsed Time** is the difference between the Start Time and the End Time.
- **Job Type** displays the type of task that is being performed, such as Create Application or Update Dimensions.
- **Status** displays the current state of the job, such as Running, Success or Failure.
- **Comment** shows a user-entered note or details about a specific job. The Comment is entered when the task is submitted.

4 Optional: Under **Job Details**, review additional summary level detail about the selected job. The Job Details include a taskflow ID and may include a hyperlink that enables you to click to view more detail, such as error or impact analysis information. If a hyperlink is offered, you can save the information to a file.

**Note:** **Task Flow ID** is the system-generated task ID for the specific task, displayed in the format `<application name>:\<task name>\<generated taskflow number>`.

For example, the generated taskflow number may be displayed as *AppMgmt_DeployApplication_D20120824T08520_5ed*, where *AppMgmt* is the application name, *DeployApplication* is the task, and *D20120824T08520_5ed* is the generated taskflow instance ID.

5 Optional: If you used a **View** option or the **Filter** button to show the Query by Example boxes, use the text box at the top of each column to locate a job by entering text to match in the box.

6 Optional: If required, click **Stop** to end the selected task in Running state.

**Caution!** While the task flow stops quickly after the button is pressed, activities that affect results may require additional time to ensure that the state of the data is consistent.
Preparing Templates and Flat Files for Creating and Updating Management Ledger Profitability Applications

Subtopics
- Exporting Templates
- Preparing Flat Files for Each Management Ledger Dimension
- About Flat File Properties
- About Comments in Flat Files
- Flat File Sample

You can create Management Ledger applications using the Profitability Applications Console by importing an exported template. You can also add or update dimensions in Management Ledger applications with Deploy Origin of File type by importing flat files.

The listed topics describe how to prepare templates and flat files for creating and updating Management Ledger applications.

Exporting Templates

Template files are created by exporting an entire Management Ledger application — including application metadata, dimension metadata, and program artifacts — in a single operation to create one application “template” file. Exporting and importing template files is useful for backing up applications or migrating them to another environment, from test to production, for example.

To export a template file:

1. Display the Profitability Applications Console and select a Management Ledger application.
2. Select Actions, and then Export Template.
3. Review the file name and change it if you want. Extension .zip is added to exported template files.
4. If POV data are displayed, select whether to export all POV data or clear checkboxes for data that shouldn’t be exported.
5. Click OK to start the export.

The file is created in an import_export folder on the server. Note that this folder is the same one used for LCM (lifecycle management), such as <MIDDLEWARE_HOME>/user_projects/epmsystem1/import_export.

6. You can click the Job Library tab, , and then Refresh to check export status.

Once created, a template file can be imported to create a new Management Ledger application as described in “Creating Management Ledger Applications by Importing Template Files” on page 203.
Preparing Flat Files for Each Management Ledger Dimension

You can use dimension flat files to update Management Ledger applications as well as create them. If existing members are omitted from the file, they are removed during the update. See the topics listed at the beginning of this section for details.

To add or replace dimensions in a Management Ledger application using flat files, prepare the flat files as follows:

1. In a text editor, create one flat file for each dimension, following the format described in this section. Each flat file must completely define the dimension. To import the files, see “Creating Profitability and Cost Management Applications in the Profitability Applications Console” on page 201. Dimension flat files are text files that contain the following:
   - A header record that specifies the order of dimension and member properties
   - A dimension data record with values for those properties
   - Individual member data records with values for those properties

   **Note:** Because member entries need to be in the same order as the final outline, parent members must be defined before their children. The order of the members in the file will be the same order as the dimension tree is displayed within Profitability and Cost Management (for example, in the Member Selector).

2. Create dimension and member header records following these rules:
   - You can specify the properties in any order, separated by commas. Each value of a multi-value property must be enclosed in quotes. Quotes are needed when a property has multiple values. For example, a given member of the user-defined attribute (UDA) property can have multiple UDA strings. They are enclosed in quotes to indicate they are a set, such as:
     "myUDA1, myUDA2, myUDA3",
   - Commas used within multi-value properties, such as attribute names or UDAs, are always interpreted as value separators. Single-value properties can contain commas as long as the value is enclosed in quotes (for example, "my,value").
   - To put double quotes in a value, enclose the double-quoted value in another set of quotes. For example, to enter the value "myValue", specify it as follows: ""myValue"

   **Note:** For best results, avoid using member names that require double quotes.

   - Property names are not case-sensitive: For example, the following names are all handled the same: name, NAME, Name.
   - Not all properties are required for every row in the flat file. Properties can be ignored or null, indicated by no value between the commas where that property would ordinarily be listed, such as: , ,

For example, in a member record, any dimension properties from the header would be ignored.
Properties without a value or with an invalid value will be defaulted for required properties (see “About Flat File Properties” on page 212).

Comments are supported (see “About Comments in Flat Files” on page 213).

The Rule and Balance dimensions are seeded in the same way they are seeded when deploying from a Master Cube. You do not need to import a file for them; members are created automatically. You cannot specify aliases for any Rule dimension members. The dimension member names for Rule and Balance dimension members are defined programmatically and translated.

The dimension sort and solve order precedence is handled as follows: Processing is based on alphabetical order using Dimension Name, except that Attribute dimensions are always last (and ordered alphabetically from there).

### About Flat File Properties

Table 49 on page 212 describes the format of each property in a dimension flat file. For an example of a flat file, see “Flat File Sample” on page 214.

**Note:** The dimension name is given in the first row after any comment rows.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Property Type</th>
<th>Dimension Types</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimension Type</td>
<td>Dimension</td>
<td>All</td>
<td>Generic</td>
<td>Identifies the dimension type. Valid examples are: Account, Period, Year, Version, Scenario, POV1, POV2, POV3, POV4, Dimension, Attribute, Rule. If there is no property name from this list the default will be “Generic”. This position represents the Dimension Name (if Gen1 row) or Member Name. POV orders are as follows: Year/POV1 = 1, Period/POV2 = 2, Scenario/POV3 = 3, Version/POV4 = 4.</td>
</tr>
<tr>
<td>Storage Type</td>
<td>Dimension</td>
<td>Generic, Account, Entity, POV</td>
<td>Sparse</td>
<td>Sparse and Dense</td>
</tr>
<tr>
<td>Dimension Name (as Attribute)</td>
<td>Dimension, Member</td>
<td>Generic</td>
<td>none</td>
<td>When you provide an Attribute Dimension Name in the header, that position represents an associated attribute dimension for the dimension being loaded. Any String in that position is treated as an “attribute member”. For example, suppose you specify “My Attribute Dimension” in the header. In the data rows of that file, a String in the same position is assumed to be a member of that attribute dimension. So, you could specify “My Attribute Member1” on the data row for “All Products” and the loader would assign that member to “All Products” as an attribute association.</td>
</tr>
<tr>
<td>Comment</td>
<td>Dimension, Member</td>
<td>Generic, Account, Entity, Country, POV</td>
<td>none</td>
<td></td>
</tr>
</tbody>
</table>

212
<table>
<thead>
<tr>
<th>Property Name</th>
<th>Property Type</th>
<th>Dimension Types</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hierarchy Type</td>
<td>Dimension, Member</td>
<td>Generic, Account, Entity, Country, POV</td>
<td>Stored</td>
<td>Enabled, Stored, Dynamic, Disabled</td>
</tr>
<tr>
<td>BSO Data Storage</td>
<td>Dimension, Member</td>
<td>Generic, Account, Entity, Country, POV</td>
<td>Never Share</td>
<td>Label Only, Store, Never Share, Shared, Dynamic Calc And Store, Dynamic Calc</td>
</tr>
<tr>
<td>ASO Data Storage</td>
<td>Dimension, Member</td>
<td>Generic, Account, Entity, Country, POV</td>
<td>Never Share</td>
<td>Label Only, Store, Never Share, Shared</td>
</tr>
<tr>
<td>Two Pass Calculation</td>
<td>Dimension, Member</td>
<td>Generic, Account, Entity, Country, POV</td>
<td>N</td>
<td>True or False are acceptable values (N or Y).</td>
</tr>
<tr>
<td>ASO Dimension Formula</td>
<td>Dimension, Member</td>
<td>Generic, Account, Entity, Country, POV</td>
<td>none</td>
<td></td>
</tr>
<tr>
<td>Consolidation Type</td>
<td>Member</td>
<td>Generic, Account, Entity, Country</td>
<td>Not Used</td>
<td>Add, Subtract, Multiply, Divide, Percent, Ignore, Not Used</td>
</tr>
<tr>
<td>UDA</td>
<td>Dimension, Member</td>
<td>Generic, Account, Entity, Country</td>
<td>none</td>
<td>Single text value or a list of text values (enclosed by comma). When Dimension, the value is a Dimension Name of Type = UDA. When Member, it is a Member of the specified UDA dimension. For example: myUda1 “myUda1,myUda2”</td>
</tr>
<tr>
<td>Parent</td>
<td>Member</td>
<td>Generic, Account, Entity, Country, Attribute</td>
<td></td>
<td>Identifies the parent member name. If null it means the member is Generation 2. Order matters; a referenced parent must be defined previously.</td>
</tr>
<tr>
<td>Alias:Alias table</td>
<td>Dimension, Member</td>
<td>All</td>
<td></td>
<td>Examples: “Alias: Default”, “Alias: T1”</td>
</tr>
<tr>
<td>Description</td>
<td>Dimension, Member</td>
<td>All</td>
<td></td>
<td>Optional – no default.</td>
</tr>
</tbody>
</table>

**About Comments in Flat Files**

For single line comments, place the hash character as the first character on the line. Blank lines are ignored.

Block comments are delineated by a start comment block indicator (`#!`) and terminated on a separate line with an end block indicator (`#--`). Intervening lines need not be commented.

For example:

```
#!—start of comment block
Comment within block
```
Administrators or other users with appropriate security provisioning can create dimensions and dimension members in an Essbase Master Cube. These are then imported into a Profitability and Cost Management application in Profitability Applications Console. The Master Cube is an ASO Application. The same Master Cube can be used to create multiple Profitability and Cost Management applications. The application type — Standard, Detailed, or Management Ledger — is set in the Console when you create the application. It does not come from the Essbase Master Cube.

This section describes how to create Essbase Master Cubes for use with the Profitability Applications Console. To use a flat file instead, see “Preparing Templates and Flat Files for Creating and Updating Management Ledger Profitability Applications” on page 210.
To create the Essbase Master Cube:

1 Create the Essbase Master database for your Profitability and Cost Management application, using the following guidelines to generate the dimensions and members for the application:
   - “Dimension Types” on page 215
   - “ASO Dimension and Member Properties” on page 217
   - “Specifying BSO and Member Dimension Properties” on page 218

For detailed instructions on creating applications and databases in Essbase, see the Oracle Essbase Database Administrator’s Guide.

Caution! The Essbase Master database name must be the same as the Essbase Master Cube name. If these names differ, users cannot update dimensions when deploying using the Profitability Applications Console.

2 Ensure that users who will create the Profitability and Cost Management application have been provisioned as an Administrator or other user with application creation provisioning.

3 Create the Profitability and Cost Management application, as outlined in “Creating Profitability and Cost Management Applications in the Profitability Applications Console” on page 201.

Caution! If you rename or delete dimensions in the Essbase Master Cube, you will no longer be able to update dimensions properly on already deployed Profitability and Cost Management applications that reference those dimensions. In other words, you should not modify in any way the UDAs that define the dimension type. If you want to rename or delete dimensions, consider creating a separate Essbase Master database so that your existing Profitability and Cost Management applications can still be updated if necessary. The existing validations in the Pre-Update Analysis option do not report these cases.

Note: When deploying the Essbase Master Cube, the ASO dimension members and their properties, such as ASO Storage Type, ASO Formula, Hierarchy Type, and so on, are automatically read into Essbase. No manual actions are required.

BSO-specific dimension and member properties cannot be read automatically during deployment. To manage this, you must create User-Defined Attribute Dimensions (UDAs) for BSO dimension members and properties, such as Formulas, Data Storage and Dimension Storage type, and specific Performance Management Architect dimensions and properties, such as POV dimensions, to align the BSO database with Essbase.

The Unicode property set for the Essbase Master Cube dictates the Unicode property for any Reporting or Calculation database applications that are created from a Profitability and Cost Management application deployed from that Master Cube.

**Dimension Types**

When creating the Essbase Master Cube, every dimension must be assigned a DIMTYPE UDA.
The valid dimension types for Standard Profitability applications are displayed in Table 50, “Standard Profitability Dimension Types”.

<table>
<thead>
<tr>
<th>Property Label</th>
<th>Property Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| Allocation Type| DIMTYPE_ALLOCTYPE | For Standard Profitability only, create an AllocationType dimension using the following criteria:  
  - Assign the Generation1 member a UDA value of “DIMTYPE_ALLOCTYPE” so that Profitability and Cost Management recognizes the member. The member name may be anything, but the UDA value must be set as described.  
  - Create one dummy member in this dimension to enable Essbase to save an outline where a Gen1 member has no children. |
| Measures       | DIMTYPE_MEASURES  |  
  - Create a Measures dimension and assign the Generation1 member a UDA value of “DIMTYPE_MEASURES.” The member name may be anything, but the UDA value must be set as described.  
  - Create one dummy member in this dimension to enable Essbase to save an outline where a Gen1 member has no children.  
  **Note:** For Standard Profitability only, see “Custom Measures Dimensions” on page 221 to use Custom Measures dimensions. |
| Generic (Business) | DIMTYPE_GENERIC | Assign Business dimensions a UDA value of DIMTYPE_GENERIC.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| POV            | DIMTYPE_POV1, DIMTYPE_POV2, DIMTYPE_POV3, DIMTYPE_POV4 | For up to four POV dimensions, assign the Generation1 Member value in numerical sequence. For example, DIMTYPE_POV1, DIMTYPE_POV2, and so on.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |

Dimension types for Detailed and Management Ledger Profitability applications are similar except Detailed Profitability has the system dimension MeasuresDetailed (DIMTYPE_MEASURES) and Management Ledger Profitability has system dimensions Rule (DIMTYPE_RULES) and Balance (DIMTYPE_BALANCES).

**User-Defined Attributes**

User-defined attributes (UDAs) enable analysis based on text attributes of dimension members. For example, you can add a UDA called “New Products.”

Enter the UDA name for the selected member.

You can enter up to 256 alphanumeric characters. A limit of 80 characters is applied for Unicode-enabled databases.

Attribute dimensions in the Master Cube must have dimension names composed of ASCII characters only (English letters and numbers), and the first character of the attribute dimension name must be a letter, for example, a-z or A-Z.
In any Essbase outline, ensure all dimensions have a unique name. Otherwise, the creation of the outline will fail. For example, an attribute dimension member name cannot match a regular dimension name.

**ASO Dimension and Member Properties**

The properties for the ASO dimension and member properties displayed in Table 51, “Profitability and Cost Management ASO Dimension and Member Properties” are pulled automatically into Essbase during deployment. No manual actions are required to align these properties with the Master Essbase Application.

<table>
<thead>
<tr>
<th>Property Label</th>
<th>Property Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>Alias</td>
<td>The alias is the dimension member name that is displayed in a deployed application. Any member aliases are displayed, except for the Root Member.</td>
</tr>
<tr>
<td>Comment</td>
<td>Comment</td>
<td>A comment may be entered for the dimension or member. Comments can contain up to 255 characters. By default, this text box displays the current comment, if one exists.</td>
</tr>
<tr>
<td>Consolidation</td>
<td>Consolidation</td>
<td>Member consolidation properties determine how children roll up into their parents. If the current member is not a dimension or an attribute, one of the following consolidation operators is assigned to the member:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- + (addition) - Default</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- - (subtraction)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- * (multiplication)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- / (division)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- % (percent)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ~ (ignore during consolidation)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ^ (never consolidate)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- NotUsed</td>
</tr>
</tbody>
</table>

**Note:** Some restrictions exist regarding the use of consolidation operators in aggregate storage outlines (ASO). See the Oracle Essbase Database Administrator's Guide.

<table>
<thead>
<tr>
<th>Data Storage(ASO)</th>
<th>ASODimensionDataStorage (for dimension root member)</th>
<th>The ASO storage options that are available for dimensions and members:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ASOMemberDataStorage (for dimension members)</td>
<td>StoreData—Data is stored with the dimension.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ShareData—Data associated with this member can be shared. The ShareData property applies to the member only. The Dimension Root Member cannot be shared.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> This option does not apply to stored hierarchies in aggregate storage outlines.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LabelOnly—No data is associated with this dimension.</td>
</tr>
<tr>
<td>Property Label</td>
<td>Property Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Dimension Formula (ASO)</td>
<td>ASODimensionFormula (for dimension root member)</td>
<td>The calculation formula that is applied to the dimension.</td>
</tr>
<tr>
<td>Dimension Solve Order</td>
<td>DimensionSolveOrder</td>
<td>The numeric value in the solution sequence for selected dimensions. For example, if this dimension is to be solved second, enter “2.”</td>
</tr>
</tbody>
</table>
| Hierarchy Type (Dimensions Only) | DimensionHierarchyType | The Hierarchy Type property applies only to Aggregate Storage (ASO) databases in Essbase. The type of hierarchy for the dimension is set to one of the following values:  
  - **STORED** - for any dimension members that use the following consolidation symbols:  
    - +ADDITION  
    - ~ IGNORE  
    - **Note:** The no-consolidation or IGNORE (~) operator can only be used in a STORED hierarchy if the member’s parent is set to LABEL_ONLY  
  - **DYNAMIC** - for dimension members that use any consolidation symbol, including ADDITION and IGNORE, or if the dimension member contains a formula.  
  - **ENABLED** - to support alternate hierarchies |
| Hierarchy Type (Members Only) | HierarchyType | The type of hierarchy available for the member:  
  - **STORED** (For first Gen2 child)  
  - **Dynamic** (For Second Gen2 child onwards. The Second Gen2 member can host the alternate hierarchies)  
  - **None** (Do Not Use) |
| Member Formula (ASO) | ASOMemberFormula | The calculation formula that is applied to the member. |
| Member Solve Order (Members Only) | MemberSolveOrder | Enter the numeric value in the solution sequence for selected member. For example, if this member is to be solved second, enter “2.”  
This property applies to ASO databases only.  
Members that have a solve order of 0 inherit the solve order of their dimension.  
Members with the same solve order are evaluated in the order in which their dimensions appear in the database outline, unless otherwise specified by the dimension sort order property.  
Members with no solve order are evaluated after members with a solve order. |
| Unicode | Unicode | Select **Unicode** to set the application to Unicode mode. Unicode-mode applications support multiple character sets. |

### Specifying BSO and Member Dimension Properties

The Essbase Master Cube is an ASO Application, and the ASO dimensions and members are automatically read during deployment. However, BSO-specific member properties are required
for BSO databases (for Standard Profitability applications, for example, or to handle special dimensions and members that cannot be read automatically).

To include these BSO and Performance Management Architect dimensions and members in the Essbase BSO database, a Generation1 member is required for each dimension to be deployed into Profitability and Cost Management. The order of the dimensions in the master outline is determined by the order of the dimensions in the Profitability and Cost Management generated outline.

You must create UDAs for each of the dimensions and members listed on Table 52, “Dimension and Member Properties for Profitability and Cost Management”. For additional information about working with Essbase, see the Oracle Essbase Database Administrator’s Guide.

**Caution!** Do not assign multiple UDA’s of the same type to a specific member in Essbase as this results in inconsistent behavior because the Profitability and Cost Management deployment does not know which UDA type to use. For example, do not assign both DIMTYPE_POV1 and DIMTYPE_POV2 to the same member.

See Table 52, “Dimension and Member Properties for Profitability and Cost Management” for the properties that require special UDAs that must be assigned manually to the application:

**Note:** Because the BSO properties are assigned using a UDA, if a user inadvertently assigns incompatible BSO properties in the Essbase Master database, a subsequent "Deploy to Reporting Cube" operation within Profitability and Cost Management may fail. If this situation occurs, you can identify the incompatible assignments by manually loading the data and rules files that were generated by the Reporting database deploy.
Table 52  Dimension and Member Properties for Profitability and Cost Management

<table>
<thead>
<tr>
<th>Property Label</th>
<th>Property Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Storage(BSO)</td>
<td>• BSODimensionDataStorage</td>
<td>For BSO Data Storage, assign one of the following UDAs to the Generation1 member of any child member:</td>
</tr>
<tr>
<td></td>
<td>(for dimension root member)</td>
<td>• BSODS_LABELONLY</td>
</tr>
<tr>
<td></td>
<td>• BSOMemberDataStorage</td>
<td>• BSODS_NEVERSHARE</td>
</tr>
<tr>
<td></td>
<td>(for dimension members)</td>
<td>• BSODS_SHAREDDATA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• BSODS_DYNCALANDSTORE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• BSODS_DYNCALC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• BSODS_STOREDATA</td>
</tr>
<tr>
<td></td>
<td>For BSO Data Storage, assign one of the following UDAs to the Generation1 member of any child member:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• BSODS_LABELONLY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• BSODS_NEVERSHARE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• BSODS_SHAREDDATA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• BSODS_DYNCALANDSTORE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• BSODS_DYNCALC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• BSODS_STOREDATA</td>
</tr>
<tr>
<td></td>
<td>When setting the UDA, if no value is assigned to the property BSO Data Storage, the Profitability and Cost Management deployment assumes the following default values are assigned:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Generation1 members are assigned “BSODS_LABELONLY”, with the following exceptions:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• POV Generation1 Members are set to “BSODS_STOREDATA”</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Measures Generation1 Member is set to “BSODS_DYNCALC”</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• All other members are assigned “BSODS_STOREDATA”</td>
</tr>
<tr>
<td>Dimension</td>
<td>BSODimensionFormula (for dimension root member)</td>
<td>For the BSO Dimension Formula, the user cannot specify the BSO Dimension formula in the Essbase Master database, because that database has to be an ASO database (and therefore the formula attribute available in the master database is the ASO formula).</td>
</tr>
<tr>
<td>Formula(BSO)</td>
<td></td>
<td>After deployment, to have the BSO formula display in the Essbase Reporting database, manually enter the BSO dimension formula directly into the Reporting database. The BSO formula will have to be reentered if you redeploy.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Use the appropriate BSO CALCULATOR syntax.</td>
</tr>
<tr>
<td>Dimension Sort</td>
<td>DimensionSortOrder</td>
<td>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.</td>
</tr>
<tr>
<td>Order</td>
<td></td>
<td>For example, if this dimension is to be the second in the Essbase outline, enter “2.”</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Dimension Sort Order must be set for every dimension in the model, except Alias and UDA dimensions.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The dimension sort order must be sequential, unique, and greater than or equal to 1.</td>
</tr>
<tr>
<td>Dimension Storage</td>
<td>DimensionStorageType</td>
<td>The Dimension Storage Type property applies only to Block Storage (BSO) databases in Essbase.</td>
</tr>
<tr>
<td>Type</td>
<td></td>
<td>By default, the value is set to “SPARSE”.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If you require a DENSE setting, assign a UDA with the value “DIMSTORETYPE_ DENSE” to the Generation1 member.</td>
</tr>
</tbody>
</table>
**Property Label** | **Property Name** | **Description**
--- | --- | ---
Member Formula (BSO) | BSOMemberFormula | For the BSO Member Formula, the user cannot specify the BSO member formula in the Essbase Master cube, because that database has to be an ASO database (and therefore, the formula attribute available in the Master Cube is the ASO formula).

After deployment, to have the BSO formula display in the Essbase Reporting database, manually enter the BSO dimension formula directly into the Reporting database. The BSO formula will have to be reentered if you redeploy.

Use the appropriate BSO CALCULATOR syntax.

Two Pass Calculation (Dimensions Only) | BSO_TWOPASS (for dimension root member) | For BSO databases only, specify the BSO_TWOPASS to calculate a member on the second pass through the outline.

Two Pass Calculation (Members Only) | BSO_TWOPASS | For BSO databases only, specify the BSO_TWOPASS to calculate a member on the second pass through the outline.

**Viewing Block Storage Option (BSO) Data Storage Values Assigned to Member Properties**

To understand what BSO data storage values are assigned to members, use the HPM_DIM_MEMBER_PROP_V view to debug when the deploy to Essbase fails for the BSO Calculation cube.

The view consists of the following columns:

APPLICATION_NAME
DIMENSION_NAME
MEMBER_NAME
BSO_DATA_STORAGE
DIMENSION_STORAGE_TYPE

When queried, the view returns a row for each dimension member in each deployed Profitability and Cost Management application.

**Custom Measures Dimensions**

For Standard Profitability only, if you want to use custom measure members, create a member name “UserDefinedDriverMeasures” under the Driver Measures, and add custom members as children of that member. The Profitability Applications Console inserts those custom members as children of the “UserDefinedDriverMeasures” in the deployed Measures hierarchy.

The member UserDefinedDriverMeasures is where the application-specific, user-defined driver measures are stored. Set the ASOMember DataStorage and BSOMember DataStorage properties as follows:
Set to **StoreData** if the member does not have children.

Set to **LabelOnly** if members are added as children to this member, and all these children have the consolidation symbols of **IGNORE**.

**Note:** All driver measures must be unique in the outline. Do not use the name of an existing driver measure in a 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.
Importing Data into Profitability and Cost Management

In This Appendix

About Staging Tables ................................................................. 224
Creating Import Configurations ............................................... 225
Modifying Import Configurations ........................................... 227
Deleting Import Configurations ............................................. 228
Running Import Configurations ............................................. 228
Verifying Imported Data .......................................................... 229

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 Standard and Detailed Profitability applications 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 Oracle Hyperion EPM Architect Dimension Library through EPM Workspace. See the chapter on importing metadata in the Oracle Hyperion Enterprise Performance Management Architect Administration 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.
- Model and end-user data can be imported and exported from Essbase for Standard Profitability applications only. See the Oracle Essbase Database Administrator’s Guide.
- Model data and applications can be imported using Lifecycle Management. See the Oracle 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.

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.
About Staging Tables

When data is imported into Profitability and Cost Management, users create the staging tables that provide the predefined structure to manage the import.

To import model data from relational databases into Profitability and Cost Management, you must create a set of staging tables in a separate database schema from the location in which the Profitability and Cost Management database tables were created to format the information for use in the application.

**Caution!** Do not create import staging tables inside the product schema. Modifications of the product schema are not only unsupported, but can produce unpredictable results.

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. Staging database scripts are available for Microsoft SQL Server and Oracle Database after installation in the installation folder. By default, the location is \%EPM\_ORACLE\_HOME\%\products\Profitability\database\Common\%. Use the appropriate script for your application type to create the staging tables in the new database:

- For Standard Profitability applications, use the create_staging.sql script.
- For Detailed Profitability applications, use the create_dp_staging.sql script.

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 (For Standard Profitability only)
- POV
- Driver
- Driver Selection
- Driver Exception
- Assignment (For Standard Profitability only)
- Source Assignment Rule Associations
- Calculation Rules (For Detailed Profitability only)

The staging tables are created by the Profitability and Cost Management administrator (admin), using the formats specified in these sections:

- Chapter 6, “Standard Profitability Import Staging Tables”
- Chapter 15, “Importing Detailed Profitability Staging Tables”

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

Data can be imported only if all of the following elements 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, Driver Exceptions, Assignments, Assignment Rules Selections, and Calculation Rules.

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.

   **Caution!** Oracle recommends that, before importing data or artifacts, you create a backup directory of your databases in EPM Workspace and Essbase. Contact your administrator for assistance.

2. **Run the appropriate script against the source database to automatically create the staging tables:**
   
   - For Standard Profitability applications, run the `create_staging.sql` script. The tables are generated using the schema provided in Chapter 6, “Standard Profitability Import Staging Tables”.
   
   - For Detailed Profitability applications, run the `create_dp_staging.sql` script. The tables are generated using the schema provided in Chapter 15, “Importing Detailed Profitability Staging Tables”.

3. **Load the model data into the source staging tables.**

4. **Review the entries in the staging tables for obvious issues, and remove 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.**
From **Task Areas**, select **Manage Model**, then **Import Staging Tables**.

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

Step 1 of the Import Data dialog box is displayed.

Under **Data Source Details**, 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.

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

Click **Next**. Step 2 of the Import Data dialog box is displayed.

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

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

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.
Click Next.

Step 3 of the Import Data dialog box is displayed.

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.

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.

Optional: From the Import Configurations List, select the import configuration, and click the Run import configuration button to run the import configuration. See “Running Import Configurations” on page 228.

Modifying Import Configurations

When updating an import configuration, the target application, set of staging tables and import configuration name can be changed.

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 the import configuration to be modified.

3. Click the Edit import configuration 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.

6. Optional: Under Staging Tables Details, change the set of staging tables to be imported.

7. Click Next.

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

9. Click Finish.

The modified configuration is stored, and can be run at any time. See “Running Import Configurations” on page 228.
Deleting Import Configurations

To delete import configurations:

1. From Task Areas, select Manage Model, then Import Staging Tables.
   The Import page is displayed.

2. Select the configuration to be deleted, and click the Delete import configuration button.
   A message asks you to confirm the deletion.

3. 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. Click the Run import configuration button.
   The Import Data dialog box is displayed. All data connection details are listed.

5. Click Next.

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

7. Click Next.
   The Configuration Summary is displayed. The Configuration Name and connection URI are identified.

8. 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.

9. 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.

10 **When the import is complete, verify the imported data.** See “Verifying Imported Data” on page 229.

### 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 Hyperion Enterprise Performance Management Workspace, select Navigate, then Applications, then Profitability, and then the application 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 errors or warnings.**
  4. **Correct errors in the source staging tables, and then rerun the import.**
     Repeat step 2 through step 3 as required until no errors are generated during the import.
About Lifecycle Management

Lifecycle Management provides a consistent way for Oracle Enterprise Performance Management System products to migrate an application, a repository, or individual artifacts across product environments and operating systems.

Using Lifecycle Management features, you can perform the following tasks:

- View applications and folders
- Search for artifacts
- Migrate directly from one application to another
- Migrate to and from the file system
- Save and load migration definition files
- View selected artifacts
- Audit migrations
- View the status of migrations
- Importing and exporting individual artifacts for quick changes on the file system

Generally, the Lifecycle Management interface in Shared Services Console is consistent for all EPM System products that support Lifecycle Management; however, Oracle Enterprise Performance Management System products display different artifact listings and export and import options in the Lifecycle Management interface.

In addition to providing the Lifecycle Management interface in Oracle 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.

Lastly, there is a Lifecycle Management Application Programming Interface (API) that enables users to customize and extend the Lifecycle Management functionality.
For detailed information about Lifecycle Management and Profitability and Cost Management artifacts for both Standard and Detailed Profitability, see the Oracle Enterprise Performance Management System Lifecycle Management Guide.

**Modifying Default Timeout Settings for Lifecycle Management**

If you are using Oracle Hyperion Enterprise Performance Management System Lifecycle Management to import large models, the import may take longer to process than the time specified in the default timeout settings on the Oracle WebLogic Server. To work around this issue, you must reset the default timeout settings.

To modify the default timeout for WebLogic Server:

1. **Navigate to**
   
   ```
   %Middleware_HOME%\user_projects\epmsystem1\httpConfig\ohs\config\OHS\ohs_component\mod_wl_ohs.conf.
   ```

2. **In the configuration file, locate the section LocationMatch/profitability:**
   
   ```
   <LocationMatch /profitability>
   SetHandler weblogic-handler
   PathTrim /
   KeepAliveEnabled ON
   KeepAliveSecs 20
   WLIOTimeoutSecs 3000
   Idempotent OFF
   WeblogicCluster servername:6756
   </LocationMatch>
   ```

3. **Add the following lines within the LocationMatch/profitability section:**
   
   ```
   WLIOTimeoutSecs 3000
   Idempotent OFF
   ```

4. **Navigate to**
   
   ```
   %Middleware_HOME%\user_projects\epmsystem1\httpConfig\ohs\config\OHS\ohs_component\httpd.conf
   ```

5. **Set Timeout to 3000,** as shown in the following text:
   
   ```
   # Timeout: The number of seconds before receives and sends time out.
   Timeout 3000
   ```
Note: The server timeout shown above is a suggested limit, and may be modified to suit the specific timeout settings provided in the documentation for your application server.
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.

**Table 53, “Profitability and Cost Management Database Backup Requirements”,** lists the databases that must be backed up, their type, and suggested backup options.

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

<table>
<thead>
<tr>
<th>Database</th>
<th>Database Type</th>
<th>Backup Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Import Staging Area</td>
<td>Relational Database (RDB)</td>
<td>Standard backup techniques for any RDB can be used, including the following options:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Database scripting</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Scheduler scripts</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Backup tools, such as TOAD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Backup procedure from SQL Server or Oracle.</td>
</tr>
<tr>
<td>Operational Data Store</td>
<td>Relational Database (RDB)</td>
<td>Standard backup techniques for any RDB can be used, including the following options:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Database scripting</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Scheduler scripts</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Backup tools, such as TOAD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Backup procedure from SQL Server or Oracle.</td>
</tr>
<tr>
<td>Database Type</td>
<td>Backup Options</td>
<td></td>
</tr>
<tr>
<td>---------------</td>
<td>----------------</td>
<td></td>
</tr>
<tr>
<td>Block Storage Option (BSO) and Aggregated Storage Option (ASO) Database and Assignment Rules for Standard Profitability only</td>
<td>Standard Essbase backup techniques should be used. For detailed backup and recovery procedures, see the Oracle Essbase Database Administrator's Guide.</td>
<td></td>
</tr>
</tbody>
</table>
| Model Data Schema for Detailed Profitability Relational Database (RDB) | Standard backup techniques for any RDB can be used, including the following options:  
  - Database scripting  
  - Scheduler scripts  
  - Backup tools, such as TOAD  
  - Backup procedure from SQL Server or Oracle. |

For detailed backup and recovery procedures, see the Oracle Enterprise Performance Management System Backup and Recovery Guide.
Generated Calculation Script Naming Conventions

When Oracle Hyperion Profitability and Cost Management generates Essbase scripts, the scripts are automatically named using specific conventions. The driver calculation scripts and allocation calculation scripts are combined into one script per stage. The script name is created in the following format:

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

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

<table>
<thead>
<tr>
<th>Script Type</th>
<th>Cost Layer</th>
<th>Revenue Layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interstage Allocation</td>
<td>&quot;a&quot;</td>
<td>&quot;r&quot;</td>
</tr>
<tr>
<td>Intrastage Allocation</td>
<td>&quot;i&quot;</td>
<td>&quot;t&quot;</td>
</tr>
</tbody>
</table>

- The POV-identifier is based on the POV ID and may include up to three digits. A 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 because of script splitting, the _index displays the numerical sequence of the scripts for the same type, POV, Stage, and Layer, starting with 001, 002, and so on.

Sample Script Names
- `a3682001.csc` represents a calculation script for POV identifier 368, Source Stage 2, and the cost layer.
t4533002.csc represents an intrastage 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:

- For non-Unicode-mode application and database names, use no more than 8 bytes.
- For Unicode-mode application and database names, use no more than 30 characters.
- Do not use spaces 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 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 dimension or member names, aliases, and descriptions.
- Do not use quotation marks, periods, brackets, backslashes, or tabs within a name.

**Caution!** Brackets are permitted but not recommended in block storage outlines because they cause errors when converting to aggregate storage outlines.

- Duplicate member names or aliases are not allowed within the same dimension.
- 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 Essbase.
- Do not use forward slashes in member names.
- For time periods in custom calendars, do not use spaces in prefixes.
- Do not use 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 the following words:
  - ALL
  - AND
  - ASSIGN
  - AVERAGE
  - CALC
  - CALCMBR
  - COPYFORWARD
  - CROSSDIM
  - CURMBRNAME
  - DIM
  - DIMNAME
  - DIV
  - DYNAMIC
  - EMPTYPARAM
o EQ
o EQOP
o EXCEPT
o EXP
o EXPERROR
o FLOAT
o FUNCTION
o GE
o GEN
o GENRANGE
o GROUP
o GT
o ID
o IDERROR
o INTEGER
o LE
o LEVELRANGE
o LOOPBLOCK
o LOOPPARMS
o LT
o MBR
o MBRNAME
o MBRONLY
o MINUS
o MISSING
o MUL
o MULOP
o NE
o NON
o NONINPUT
o NOT
o OR
o PAREN
o PARENParm
o PERCENT
Essbase Naming Conventions for Attribute Calculations Dimension

The members that Essbase creates in the attribute calculations dimension (Sum, Count, Min, Max, and Avg) are not considered reserved words because you can change these names in the attribute calculations dimension and then use the standard name in an attribute or standard dimension.

If the outline is tagged as a unique member outline, avoid using Sum, Count, Min, Max, and Avg as member names. For example, if you use Max in a standard dimension and then create an attribute dimension, in which Essbase creates the Max member in the attribute calculations dimension, Essbase detects a duplicate name and returns the following error message:

"Analytic Server Error(1060115): Attribute Calculations dimension/member name already used."

If the outline is tagged as a duplicate member outline, and an attribute dimension (and, therefore, the attribute calculations dimension) exists before you use Sum, Count, Min, Max, and Avg as a base member, Oracle Essbase allows the duplicate name. However, if, you use Sum, Count, Min, Max, and Avg as a base member before creating an attribute dimension, the duplicate name is not allowed.