Oracle® Hyperion Profitability and Cost Management

Administrator's Guide

11.2.19
F91323-02
October 2024

Oracle Hyperion Profitability and Cost Management Administrator's Guide, 11.2.19

F91323-02

Primary Author: EPM Information Development Team

Contents

12 Calculating Detailed Profitability Models

13 Importing Detailed Profitability Staging Tables

14 Exporting Model Definition Data for Detailed Profitability

A Application Design Best Practices

B Importing Data into Profitability and Cost Management

C Migrating Data Using EPM System Lifecycle Management

D Backing Up Profitability and Cost Management Components

E Essbase Naming Conventions

Documentation Accessibility

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

Access to Oracle Support

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

Documentation Feedback

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

Part I

Getting Started with Profitability and Cost Management

Related Topics

- Overview
- Managing Security and Authorizing Users

1

Overview

Related Topics

- About Profitability and Cost Management
  Oracle Hyperion Profitability and Cost Management is an analytical application that is accessed from Oracle Hyperion Enterprise Performance Management Workspace.
- Accessibility in Profitability and Cost Management
  Oracle Hyperion Profitability and Cost Management provides keyboard shortcuts for the main functions.
- Profitability Types
  Oracle Hyperion Profitability and Cost Management offers three different types of applications that are used to evaluate profitability.
- Additional Product Components
  The power and flexibility of Oracle Hyperion Profitability and Cost Management are extended through the use of the products.
- Administrative Tasks
  The Oracle Hyperion Profitability and Cost Management Administrator or *admin* role enables you to perform several tasks.
- Launching Profitability and Cost Management
  Oracle Hyperion Profitability and Cost Management can only be accessed through Oracle Hyperion Enterprise Performance Management Workspace.

About Profitability and Cost Management

Oracle Hyperion Profitability and Cost Management is an analytical application that is accessed from Oracle Hyperion Enterprise Performance Management Workspace.

It 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.
- 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 Profitability Applications to build and maintain 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.

  This guide assumes that you will use the Profitability Applications Console for managing the applications and dimensions for your models. See 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, or third party products, such as Microsoft Excel.

Accessibility in Profitability and Cost Management

Oracle Hyperion 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

Oracle Hyperion Profitability and Cost Management offers three different types of applications that are used to evaluate profitability.

- Management Ledger Profitability
- Detailed Profitability

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 Oracle Essbase and scripting syntax or programming languages.

Data for Management Ledger applications is housed in both Essbase multidimensional databases and relational databases. You create applications in the Profitability Applications Console 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 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 have been 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."

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 Oracle Hyperion 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 the Profitability Applications Console, and define the business dimensions, aliases, and measures 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. The MembersDetailed dimension contains a limited set of members to process all allocations.

For more information on working with Detailed Profitability, see Understanding Detailed Profitability Product Architecture.

Additional Product Components

The power and flexibility of Oracle Hyperion Profitability and Cost Management are extended through the use of the products.

Table 1-1 Profitability and Cost Management Product Components

Product	Description
Profitability and Cost Management Profitability Applications Console	Manage dimensions and applications.
Oracle Hyperion Enterprise Performance Management Workspace	Navigate to Profitability and Cost Management and other products, and manage applications
Oracle Hyperion Shared Services	Provision users from external systems for Profitability and Cost Management Migrate Profitability and Cost Management applications
Oracle Smart View for Office	Enter and report on Profitability and Cost Management data in Microsoft Excel spreadsheets
Oracle Hyperion Financial Reporting	Create reports and charts for Web or print distribution

Administrative Tasks

The Oracle Hyperion Profitability and Cost Management Administrator or *admin* role enables you to perform several tasks.

- Create and manage user accounts using Oracle Hyperion Shared Services.
- Provision users for Shared Services authentication.

- Manage the Shared Library and Profitability and Cost Management dimensions and members in the Profitability Applications Console.
- Generate multidimensional Oracle 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 Oracle Hyperion Enterprise Performance Management Workspace.

Launching Profitability and Cost Management

Oracle Hyperion Profitability and Cost Management can only be accessed through Oracle Hyperion Enterprise Performance Management Workspace.

To access Profitability and Cost Management:

1. Ensure the following applications have been configured, and are running:
   - EPM Workspace
   - Oracle Hyperion Shared Services
   - 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.

2

Managing Security and Authorizing Users

Related Topics

- About User Setup and Provisioning
  Before working with Oracle Hyperion Profitability and Cost Management, the Administrator must set up users and groups, and assign the appropriate security role to each one.
- Assigning Security Roles
  In Oracle Hyperion Profitability and Cost Management, each user ID is assigned a security role.
- Auditing Changes in Profitability and Cost Management
  You can monitor activity and changes in your application using the Audit feature in the Oracle Hyperion Shared Services Console, and then generate audit reports detailing the results.
- Output Log Files
  Administrators can generate log files throughout Oracle Hyperion Enterprise Performance Management Workspace.
- Oracle Diagnostic Logging (ODL) Files
  The Oracle Diagnostic Logging (ODL) location for Web applications for all the configuration logging files for each Oracle Hyperion Enterprise Performance Management Workspace product are centralized.

About User Setup and Provisioning

Before working with Oracle Hyperion 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 Oracle Hyperion 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 Oracle Hyperion Enterprise Performance Management 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 **Administer**, 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, Oracle Hyperion Enterprise Performance Management System Lifecycle Management.

5. Create groups, as required. You can provision groups within groups. See the *Oracle Enterprise Performance Management System User Security Administration Guide*.
6. Log on to EPM Workspace, and open Profitability and Cost Management.
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 Oracle Hyperion Profitability and Cost Management, each user ID is assigned a security role.

- Administrator (*admin* is the default security role when you log on to Oracle Hyperion 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 Oracle Hyperion 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.

> **Caution:**
>
> If the user requires access to other products, such as an Application Creator in Oracle Hyperion Enterprise Performance Management Workspace, or importing or exporting staging tables through Oracle Hyperion Enterprise Performance Management System Lifecycle Management,, 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 Management Ledger Profitability

Table 2-1 Management Ledger Profitability and Cost Management Security Roles

Security Role	Description
Administrator (*admin*); Type of role = Power	• Create and maintain user accounts and security roles, and provision users, using Oracle Hyperion Shared Services • Generate Oracle Essbase databases • Create and manage Oracle Hyperion Profitability and Cost Management applications • Use the Profitability Applications Console to create and manage new Profitability and Cost Management applications. • Set up and maintain application preferences • Build the model database using the Profitability Applications Console 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 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; Type of role = 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; Type of role = Interactive	• View all modeling screens • Use rule sets and rules • View Rule Balancing • View Trace Allocations • Define and run queries.
View User; Type of role = Interactive	View only access for these functions: • Trace Allocations • Rule Balancing

Security Roles for Detailed Profitability

Table 2-2 Detailed Profitability and Cost Management Security Roles

Security Role	Description
Administrator (*admin*); Type of role = Administrator	• Set up and maintain application preferences • Build the model database using the Profitability Applications Console • 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 Oracle Hyperion 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 Oracle Essbase access rights to perform Essbase ASO deployment.

Table 2-2 (Cont.) Detailed Profitability and Cost Management Security Roles

Security Role	Description
Power User; Type of role = Power	• Create and maintain user accounts and security roles, and provision users, using Oracle Hyperion 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 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.

Table 2-2 (Cont.) Detailed Profitability and Cost Management Security Roles

Security Role	Description
Interactive User; Type of role = Interactive	View (Read) the following functions: • Stages • Drivers • POVs • Driver Association • Assignments • Application Preferences • Calculation Rules • Jobs Library and Status • Table Registration
View User; Type of role = Interactive	View (Read) the following functions: • Stages • Drivers • POVs • Driver Association • Assignments • Application Preferences • Jobs Library and Status • Table Registration
Manage Taskflows; Type of role = 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; Type of role = 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*.

Auditing Changes in Profitability and Cost Management

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

There are three types of audit reports available:

- Security reports
- Artifact Reports
- Config Reports

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

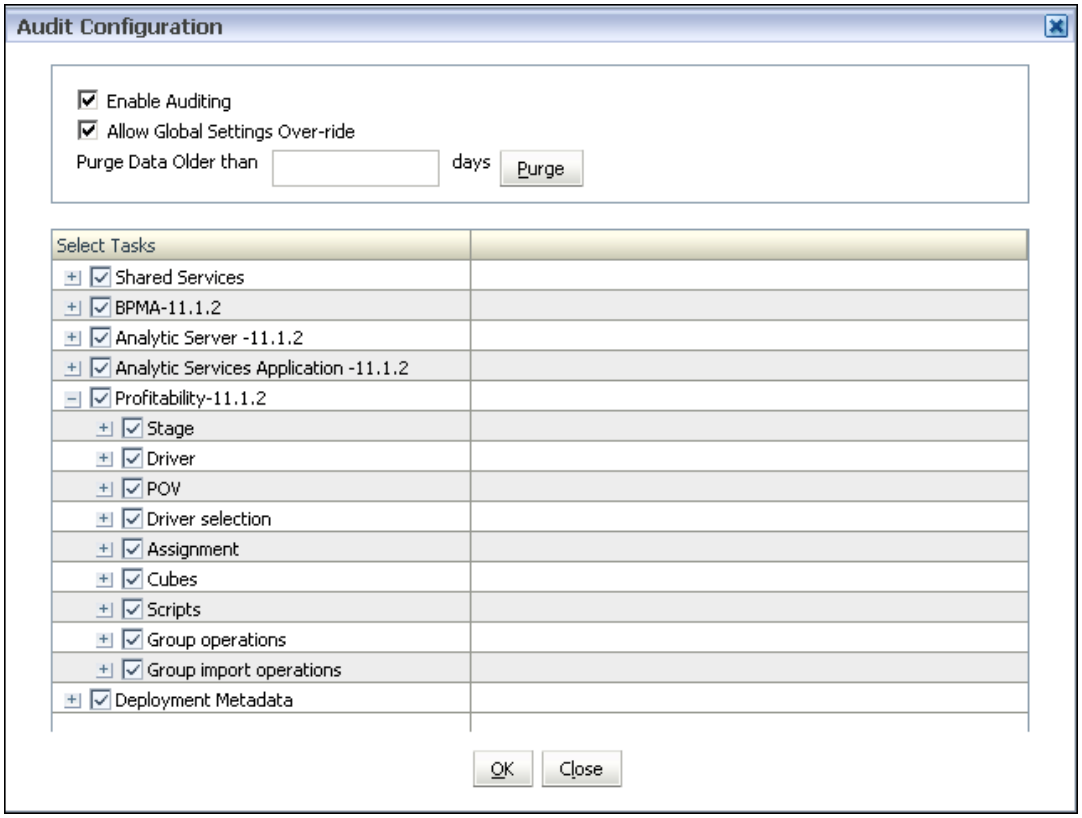
- Date
- Application
- User
- Artifact type and name
- Task that was performed

Auditing must be enabled before you can generate 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 Oracle Hyperion Enterprise Performance Management 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:**
>
> These tasks apply only to Detailed Profitability applications.

Table 2-3 Profitability and Cost Management Audit Tasks

Monitored Area	Available Monitored Tasks
Stage	• Create stage • Modify stage • Delete stage • Export stage • Import stage
Driver	• Create driver • Modify driver • Delete driver • Export driver • Import driver
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
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
Group export operations	• Export stages • Export POVs • Export drivers • Export driver selections • Export assignments

Table 2-3 (Cont.) Profitability and Cost Management Audit Tasks

Monitored Area	Available Monitored Tasks
Lifecycle Management (LCM)	• LCM Load from File • LCM Save to File • LCM Export • LCM Import

5. Click **OK**.

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

6. Click **Yes**.

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

Output Log Files

Administrators can generate log files throughout Oracle Hyperion Enterprise Performance Management Workspace.

The log files 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 Oracle Hyperion Profitability and Cost Management:

Table 2-4 Profitability and Cost Management Log Files

Log File	Description
`hpcm.log`	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\EPMSystem\servers\Profitability0\logs`. Contact your system administrator for access to this log file.
`SharedServices_Security_Client.log`	An Oracle Hyperion 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\EPMSystem\servers\Profitability0\logs`.

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 Oracle Hyperion Enterprise Performance Management Workspace product are centralized.

For Oracle Hyperion 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*.

Part II

Creating Profitability and Cost Management Applications

Related Topics

- Creating Applications Using the Profitability Applications Console

3

Creating Applications Using the Profitability Applications Console

Related Topics

- Displaying the Profitability Applications Console
  You can open the Profitability Applications console from Oracle Hyperion Enterprise Performance Management Workspace.
- Guidelines for Using the Profitability Applications Console
  The Profitability Applications Console provides several ways to manage Oracle Hyperion Profitability and Cost Management applications and dimensions.
- Creating Applications with Dimensions from an Essbase Master Cube
  You can create an application in the Profitability Applications Console using dimensions from an Essbase Master Cube.
- Creating Management Ledger Applications
  You can create Management Ledger Applications with dimensions from flat files or by importing template files.
- Working with Applications in the Profitability Applications Console
  Use the Profitability Applications Console to work with applications created from flat files and template files.
- Preparing Templates and Flat Files for Creating and Updating Management Ledger Profitability Applications
  You can prepare templates and flat files to create and update Management Ledger Profitability applications.
- Creating an Essbase Master Cube for Profitability and Cost Management
  Administrators or other users with appropriate security provisioning can create dimensions and dimension members in an Oracle Essbase Master Cube.

Displaying the Profitability Applications Console

You can open the Profitability Applications console from Oracle Hyperion Enterprise Performance Management Workspace.

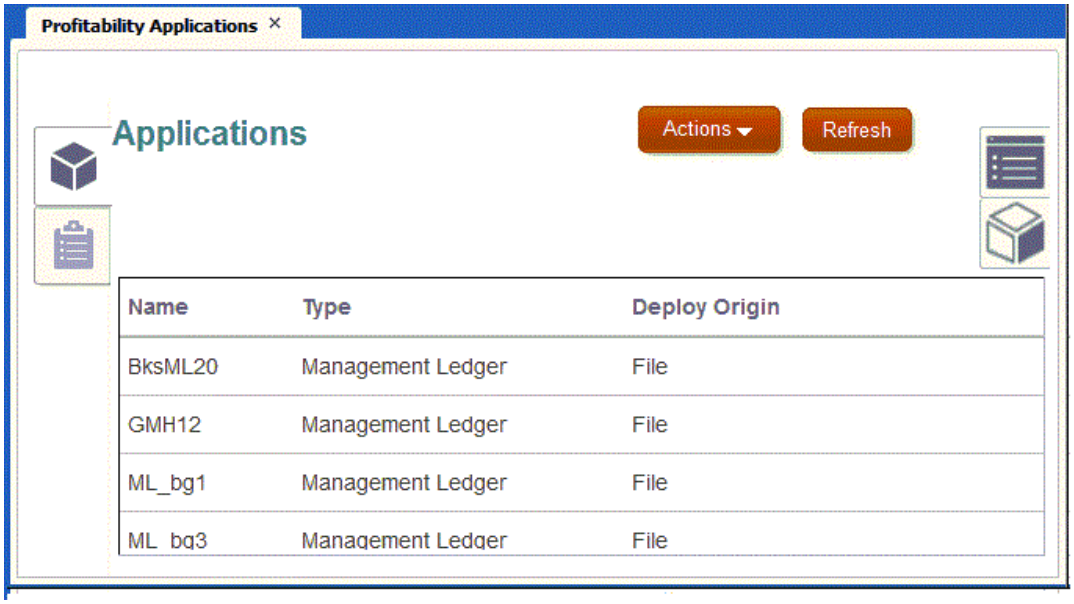
To open and view the Profitability Applications Console:

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

   The Profitability Applications Console is displayed, showing the **Applications** ( ) and **Job Library** ( ) tabs. The Profitability Applications Console lists any existing applications, their type, whether they were deployed from a Master Cube or a File, and whether they are enabled.

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

Figure 3-1 Profitability Applications Console, Applications Tab



2. Select **Actions** to perform a variety of operations on the selected application or create a new application.

   See Working with Applications in the Profitability Applications Console, and Viewing Tasks in the Profitability Applications Console Job Library.

3. Click **Refresh** to update the current tab.

> **Note:**
>
> It can take several seconds to open the Profitability Applications Console, especially after restarting the Oracle Hyperion Profitability and Cost Management service. With the Chrome and Edge Chromium browsers, this can result in a pop-up message that the pages are not responding, with an option to wait or to exit the pages. This message can be ignored – the pages will eventually open as expected.

Guidelines for Using the Profitability Applications Console

The Profitability Applications Console provides several ways to manage Oracle Hyperion Profitability and Cost Management applications and dimensions.

- Create a Profitability and Cost Management application using the **Actions, New** option with **Dimension Source** set to **Master Cube**. This type of application uses Oracle

Essbase as the source of its dimensions (Creating Applications with Dimensions from an Essbase Master Cube).

- Create 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).
- Use **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)

> **Note:**
>
> 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 **Native**). See:
>
> – Creating an Essbase Master Cube for Profitability and Cost Management
>
> – Preparing Templates and Flat Files for Creating and Updating Management Ledger Profitability Applications

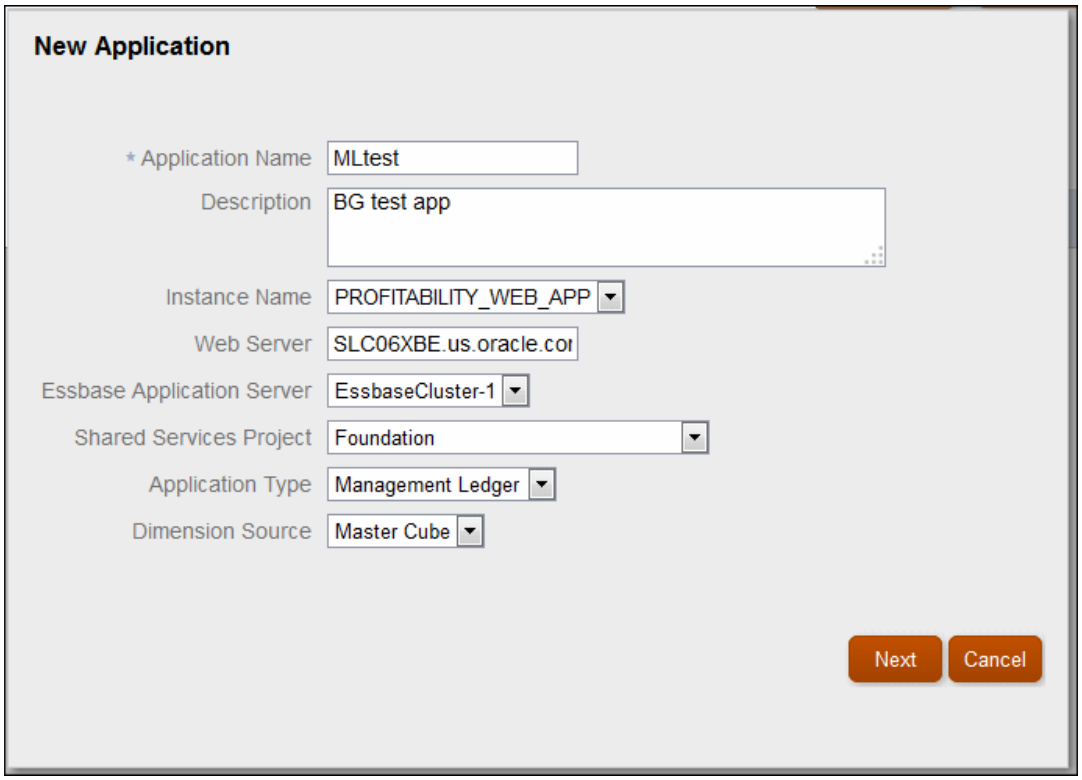
Creating Applications with Dimensions from an Essbase Master Cube

You can create an application in the Profitability Applications Console using dimensions from an Essbase Master Cube.

1. From Oracle Hyperion Enterprise Performance Management Workspace, select **Navigate**, then **Administer**, and then **Profitability Applications**.
2. Select **Actions**, and then **New**.

   The **New Application** dialog box is displayed.

Figure 3-2 The New Application Dialog Box in the Profitability Applications Console



3. In the **New Application** dialog box, enter the following information, and then click **Next:**
   - **Application Name**
   - **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 **Dimension Source**; **Master Cube** is the default
4. Select **Master Cube** as the **Dimension Source**, select the name of the Oracle 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

You can create Management Ledger Applications with dimensions from flat files or by importing template files.

> **Note:**
>
> By default, Profitability and Cost Management supports up to 15 Management Ledger applications. This limit is in place because Management Ledger applications use a dimension cache to accelerate UI interactions and improve the performance of batch jobs. During the startup of the Profitability service, the cache for all existing Management Ledger applications is loaded. The limit prevents out-of-memory errors during service startup, and long service startup times. Starting in Release 11.2.19, the applications limit is controlled by a user-configurable governor which allows you to increase this limit, provided the server is configured with sufficient memory. Contact Oracle Support if you need access to this governor.

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 Oracle Hyperion Enterprise Performance Management Workspace, select **Navigate**, then **Administer**, and then **Profitability Applications** .
2. Select **Actions**, and then **New**.
3. In the **New Application screen**, enter the following information and then select **Next:**
   - **Application Name**
   - **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 **Dimension Source**; **Master Cube** is the default and **Native** is also available for Management Ledger applications
4. Select **Native** in the **New Application** screen.

5. Enter names to use for the Rule and Balance dimensions in the new Management Ledger application. Click **Finish**. The application is added to the **Applications** tab with **Deploy Origin** of **Native**.
6. Perform the steps in Updating Application Dimensions 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).

Creating Management Ledger Applications by Importing Template Files

Exporting Templates 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.
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.

   If that file contains an input data folder, you can check **Import Input Data** to include it in the import.
5. 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

Use the Profitability Applications Console to work with applications created from flat files and template files.

Related Topics

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

About Working with Applications in the Profitability Applications Console

The **Applications** tab of the Profitability Applications Console lists Oracle Hyperion Profitability and Cost Management applications created from flat files and template files (**Deploy Origin** of type **Native**), and Oracle Essbase Master Cubes.

You can edit the selected application's **Description** and **Shared Services Project** (Editing Application Descriptions and Shared Services Projects).

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 Oracle Hyperion Enterprise Performance Management 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.

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**. See the note under **Duplicate** in step 3, following.

> **Note:**
>
> For Management Ledger applications, you can also export and import template files (Exporting Templates and Creating Management Ledger Applications by Importing Template Files)

To perform an application action in the Profitability Applications Console:

1. From Oracle Hyperion Enterprise Performance Management 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.

     > **Note:**
     >
     > When you duplicate an application in the Profitability Applications Console, dimensions and members are copied and the application is deployed. To add other artifacts such as POVs, calculation rules, and so on to Management Ledger and Detailed Profitability applications, you must export those from the source system and use Lifecycle Management (LCM) to import them into the new application (Migrating Data Using EPM System Lifecycle Management). For Management Ledger applications, you can export a template file and then import it into the new application (including input data, see Preparing Templates and Flat Files for Creating and Updating Management Ledger Profitability Applications).
   - **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 Oracle Hyperion Shared Services registration for the selected application
   - **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.
4. Respond to any confirmation prompts.

> **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).

Updating Application Dimensions

If any changes to dimensions or dimension members are required in an application with Deploy Origin of **Native** – such as adding, deleting, changing, or renaming – you must update the

dimensions in the Oracle Essbase Master Cube and then update the Oracle Hyperion 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 Oracle Hyperion Enterprise Performance Management 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 **Native**, 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 **Native**, 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.

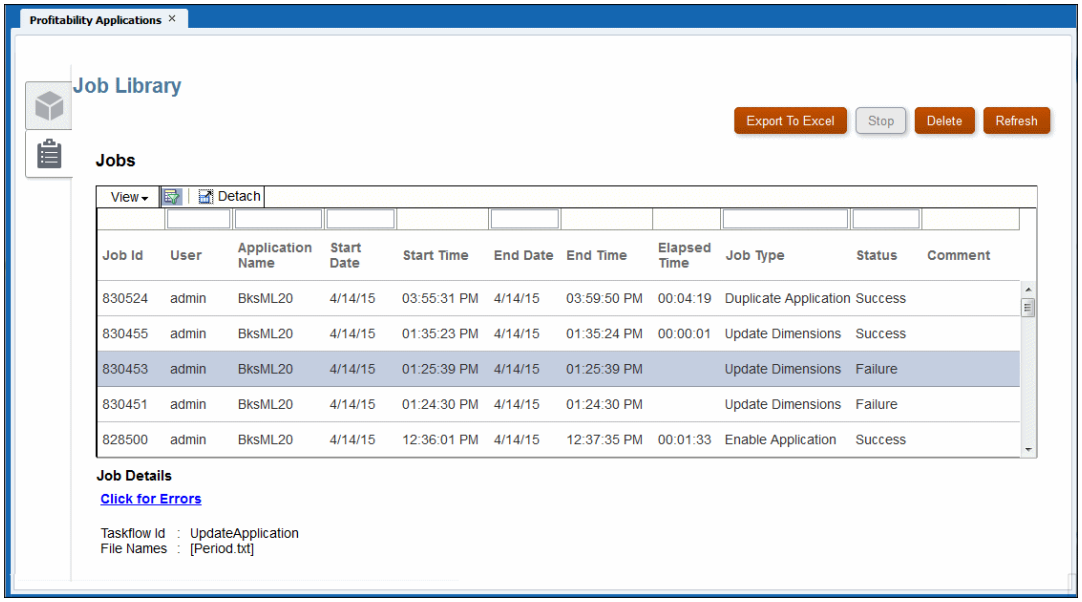
Viewing Tasks in the Profitability Applications Console Job Library

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

To view the Job Library:

1. From Oracle Hyperion Enterprise Performance Management Workspace, select **Navigate**, then **Administer**, and then **Profitability Applications**.
2. Select the **Job Library** tab ( ).

Figure 3-3 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.

> **Note:**
>
> 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.

> **Note:**
>
> The Job Library screen within an open application and the Job Library screen in the Profitability Applications Console both allow you to export the job history to an Excel spreadsheet file. The exported file is created in .html format. If you give the file an .html suffix when you save it, you can open it in any browser, in MS Word, MS Excel, or in other programs that handle .html files.

Preparing Templates and Flat Files for Creating and Updating Management Ledger Profitability Applications

You can prepare templates and flat files to create and update Management Ledger Profitability applications.

Related Topics

- About Preparing Templates and Flat Files for Creating and Updating Management Ledger Profitability Applications

About Preparing Templates and Flat Files for Creating and Updating Management Ledger Profitability Applications

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 Native type by importing flat files.

The topics listed in Preparing Templates and Flat Files for Creating and Updating Management Ledger Profitability Applications 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. **Optional:** Check **Include Input Data** to export the input data. When checked, input data for all selected POVs is exported.
6. 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`.

7. 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.

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. 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 Oracle Hyperion 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).
   - Comments are supported (see About Comments in Flat Files).
   - 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). If the Member Solve Order property is used, it overrides other considerations.

About Flat File Properties

Table 1 describes the format of each property in a dimension flat file. For an example of a flat file, see Flat File Sample.

You can include the Essbase Member Solve Order property for a member, but you must add it to the first line of the flat files to show its position.

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

Table 3-1 Properties Defined in a Dimension Flat File

Property Name	Property Type	Dimension Types	Default Value	Description
Dimension Type	Dimension	All	Generic	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
Storage Type	Dimension	Generic, Account, Entity, POV	Sparse	Sparse and Dense

Table 3-1 (Cont.) Properties Defined in a Dimension Flat File

Property Name	Property Type	Dimension Types	Default Value	Description
Dimension Name (as Attribute)	Dimension, Member	Generic	none	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.
Comment	Dimension, Member	Generic, Account, Entity, Country, POV	none	
Hierarchy Type	Dimension, Member	Generic, Account, Entity, Country, POV	Stored	Enabled, Stored, Dynamic, Disabled
BSO Data Storage	Dimension, Member	Generic, Account, Entity, Country, POV	Never Share	Label Only, Store, Never Share, Shared, Dynamic Calc And Store, Dynamic Calc
ASO Data Storage	Dimension, Member	Generic, Account, Entity, Country, POV	Never Share	Label Only, Store, Never Share, Shared
Two Pass Calculation	Dimension, Member	Generic, Account, Entity, Country, POV	N	True or False are acceptable values (N or Y).
ASO Dimension Formula	Dimension, Member	Generic, Account, Entity, Country, POV	none	

Table 3-1 (Cont.) Properties Defined in a Dimension Flat File

Property Name	Property Type	Dimension Types	Default Value	Description
Member Solve Order	Member	Generic, Account, Entity, Country	none	In Oracle Essbase, the solve order number determines the order by which members are evaluated in the dimension. You can enter a number between 1 and 127. The member with the highest solve order number is evaluated first (for example, a formula with a solve order of 20 is evaluated before a formula with a solve order number of 5). Members with the same solve order number are evaluated in the order in which their dimensions appear in the database outline. Members with no solve order number are evaluated after all members with solve order numbers.
Consolidation Type	Member	Generic, Account, Entity, Country	Not Used	Add, Subtract, Multiply, Divide, Percent, Ignore, Not Used

UDA	Dimension, Member	Generic, Account, Entity, Country	none	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"`

Parent	Member	Generic, Account, Entity, Country, POV, Attribute		Identifies the parent member name. If null it means the member is Generation 2. Order matters; a referenced parent must be defined previously.

Table 3-1 (Cont.) Properties Defined in a Dimension Flat File

Property Name	Property Type	Dimension Types	Default Value	Description
Alias:*Alias table*	Dimension, Member	All		Examples: "Alias: Default", "Alias: T1"
Description	Dimension, Member	All		Optional – no default.

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

Another comment within block

#--!
```

Flat File Sample

Figure 1 shows a sample file for dimension Customers.

Figure 3-4 Customers.txt Management Ledger Flat File Sample

```
Generic,Storage Type,Hierarchy Type,Attributes Header,comment,bso data storage,aso data storage,
    two pass calculation,aso dimension formula,consolidation type,uda,parent,alias:Default,alias:English
Customers,SPARSE,STORED,,,LABELONLY,STOREDATA,N,,,UDA,,alias:Default,alias:English
NoCustomer,SPARSE,Disabled,,,StoreData,StoreData,N,,+,,,No Customer,No Customer
AllCustomers,SPARSE,Disabled,,,StoreData,StoreData,N,,+,,,All Customers,All Customers
Big Box,SPARSE,,,,StoreData,StoreData,N,,+,,AllCustomers,,
BB100,SPARSE,,,,StoreData,StoreData,N,,+,,Big Box,Q Mart,Q Mart
BB200,SPARSE,,,,StoreData,StoreData,N,,+,,Big Box,Bike Depot,Bike Depot
BB300,SPARSE,,,,StoreData,StoreData,N,,+,,Big Box,Mountain Adventures,Mountain Adventures
Specialty Retailers,SPARSE,,,,StoreData,StoreData,N,,+,,AllCustomers,,
SR100,SPARSE,,,,StoreData,StoreData,N,,+,,Specialty Retailers,Bobs Bikes,Bobs Bikes
SR200,SPARSE,,,,StoreData,StoreData,N,,+,,Specialty Retailers,Rose Town Bikes,Rose Town Bikes
SR300,SPARSE,,,,StoreData,StoreData,N,,+,,Specialty Retailers,The Cyclery,The Cyclery
Webstore,SPARSE,,,,StoreData,StoreData,N,,+,,AllCustomers,,
```

Creating an Essbase Master Cube for Profitability and Cost Management

Administrators or other users with appropriate security provisioning can create dimensions and dimension members in an Oracle Essbase Master Cube.

These are then imported into a Oracle Hyperion 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 — 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.

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
   - ASO Dimension and Member Properties
   - Specifying BSO and Member Dimension Properties

   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.

> **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. to align the BSO database with Essbase.

> **Note:**
>
> When a Detailed application that was created using the Master Cube method is duplicated, the Profitability and Cost Management service may need to be restarted before the cube in the duplicated application can be successfully deployed to Essbase.

Dimension Types

When creating the Oracle Essbase Master Cube, every dimension must be assigned a DIMTYPE UDA.

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 80 alphanumeric characters.

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 Oracle 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 1 are pulled automatically into Oracle Essbase during deployment. No manual actions are required to align these properties with the Master Essbase Application.

Table 3-2 Profitability and Cost Management ASO Dimension and Member Properties

Property Label	Property Name	Description
Alias	Alias	The alias is the dimension member name that is displayed in a deployed application. Any member aliases are displayed, except for the Root Member.
Comment	Comment	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.
Consolidation	Consolidation	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: • + (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 (ASO). See the *Oracle Essbase Database Administrator's Guide*.
Data Storage(ASO)	• ASODimensionDataStorage (for dimension root member) • ASOMemberDataStorage (for dimension members)	The ASO storage options that are available 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.
Dimension Formula(ASO)	ASODimensionFormula (for dimension root member)	The calculation formula that is applied to the dimension.

Table 3-2 (Cont.) Profitability and Cost Management ASO Dimension and Member Properties

Property Label	Property Name	Description
Dimension Solve Order	DimensionSolveOrder	The numeric value in the solution sequence for selected dimensions. For example, if this dimension is to be solved second, enter "2."
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.

Table 3-2 (Cont.) Profitability and Cost Management ASO Dimension and Member Properties

Property Label	Property Name	Description
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.

Specifying BSO and Member Dimension Properties

The Oracle Essbase Master Cube is an ASO Application, and the ASO dimensions and members are automatically read during deployment.

To include these BSO dimensions and members in the Essbase BSO database, a Generation1 member is required for each dimension to be deployed into Oracle Hyperion 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 1. 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 1for 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 3-3 Dimension and Member Properties for Profitability and Cost Management

Property Label	Property Name	Description
Data Storage(BSO)	• BSODimensionDataStorage (for dimension root member) • BSOMemberDataStorage (for dimension members)	For **BSO Data Storage**, assign one of the following UDAs to the Generation1 member of any child member: • BSODS_LABELONLY • BSODS_NEVERSHARE • BSODS_SHAREDATA • BSODS_DYNCALANDSTORE • BSODS_DYNCALC • BSODS_STOREDATA 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: • Generation1 members are assigned "BSODS_LABELONLY", with the following exceptions: – POV Generation1 Members are set to "BSODS_STOREDATA" – Measures Generation1 Member is set to "BSODS_DYNCALC" • All other members are assigned "BSODS_STOREDATA"
Dimension Formula(BSO)	BSODimensionFormula (for dimension root member)	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). 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.

Table 3-3 (Cont.) Dimension and Member Properties for Profitability and Cost Management

Property Label	Property Name	Description
Dimension Sort Order	DimensionSortOrder	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.
Dimension Storage Type	DimensionStorageType	The **Dimension Storage Type** property applies only to Block Storage (BSO) databases in Essbase. By default, the value is set to "SPARSE". If you require a DENSE setting, assign a UDA with the value "DIMSTORETYPE_DENSE" to the Generation1 member.
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 Oracle 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 Oracle Hyperion Profitability and Cost Management application.

Custom Measures Dimensions

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.

Part III

Working with Management Ledger Profitability

Related Topics

- Understanding Management Ledger Profitability Application Architecture
- Understanding Management Ledger Dimensions
- Performing Other Application Management Tasks

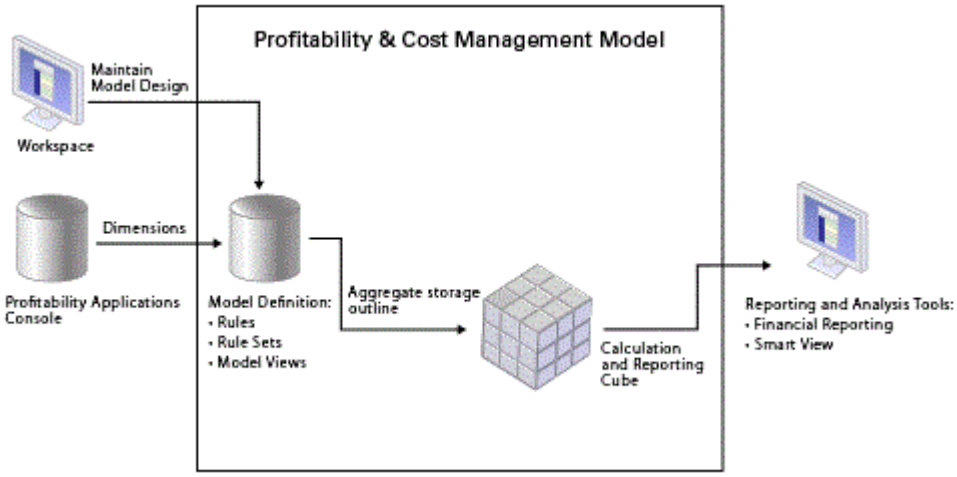
4

Understanding Management Ledger Profitability Application Architecture

Accessed through Oracle Hyperion Enterprise Performance Management Workspace, Oracle Hyperion Profitability and Cost Management is an analytical tool that resides on top of Oracle 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. For a description of Management Ledger Profitability, see Management Ledger Profitability.

Figure 4-1 Management Ledger Profitability Product Architecture



The model metadata from EPM Workspace 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 Oracle Hyperion Shared Services for the centralized management of application metadata and security.

Application administrators create the Profitability and Cost Management dimensions using the Profitability Applications Console. 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 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
- 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 and Oracle Smart View for Office.

5

Understanding Management Ledger Dimensions

Related Topics

- About Management Ledger Dimensions
  Oracle Hyperion Profitability and Cost Management uses dimensions and members created in Oracle Essbase and the Profitability Applications Console to represent many of the structural elements of the business model.
- Management Ledger System Dimensions
  Management Ledger applications must contain two system dimensions: Rule and Balance.
- Management Ledger Business Dimensions
  Business dimensions describe the business-specific objects within the model, such as products, customers, regions, and employees.
- POV Dimensions
  POV dimensions indicate a specific point of view or version of the model, such as year, scenario, or period.
- 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.
- Alias Dimensions
  Aliases are alternate names, descriptions, languages, or other items that help to define dimensions.
- Management Ledger Dimension Sort Order
  The Dimension Sort Order property controls the order of dimensions in the Oracle Essbase outline that is generated by Oracle Hyperion Profitability and Cost Management.

About Management Ledger Dimensions

Oracle Hyperion Profitability and Cost Management uses dimensions and members created in Oracle Essbase and the Profitability Applications Console 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 and other Oracle Hyperion Enterprise Performance Management 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
- Management Ledger Business Dimensions
- POV Dimensions
- Attribute Dimensions

- Alias Dimensions

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**.

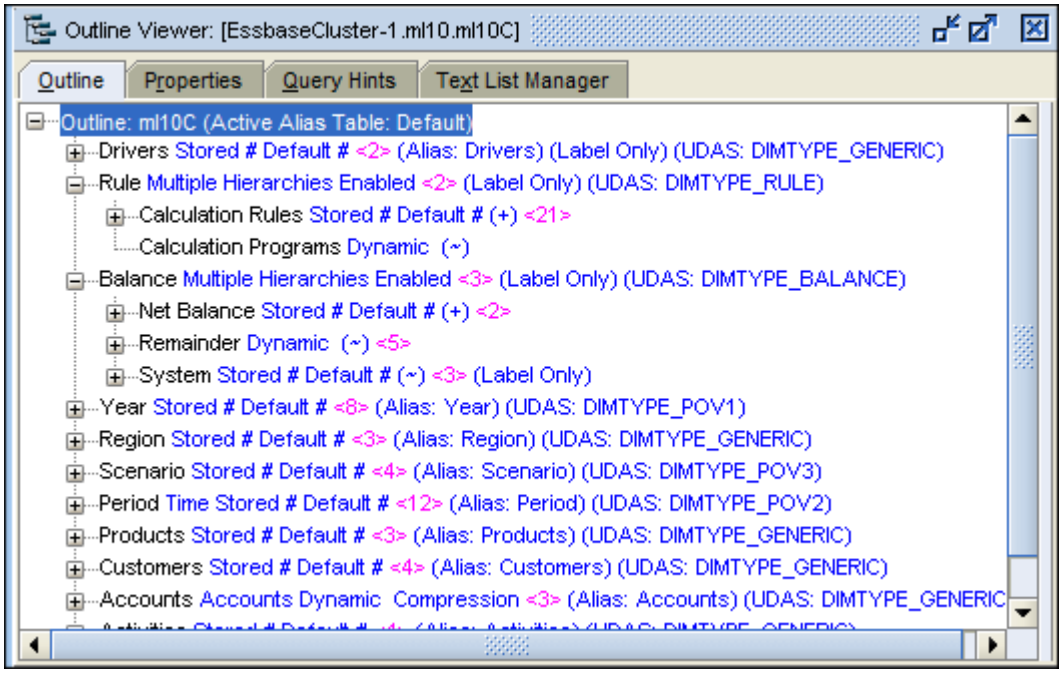
  Users can edit and add more members to the Rule dimension For example, `R1001` through `R1500`. Optionally, they can also trim that dimension. The Calculation Programs member in the Rule dimension is not editable.
- Applications must contain one and only one system dimension named **Balance**.

  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 1 shows a sample Essbase outline of a Management Ledger Profitability database, shown on the Essbase console.

Figure 5-1 Essbase Outline of a Management Ledger Profitability Database



> **Note:**
>
> For Releases 11.2.15 or later, ensure that you use the Essbase 21c Web Console Outline Screen.

Management Ledger System Dimensions

Management Ledger applications must contain two system dimensions: Rule and Balance.

These system dimensions are populated from the Profitability Applications Console when a new Management Ledger application is deployed or created. For additional information about the Rule and Balance dimensions, see the listed sections.

For detailed instructions on creating and maintaining the dimensions and members, see Creating Applications Using the Profitability Applications Consoleand the *Oracle Essbase Database Administrator's Guide*.

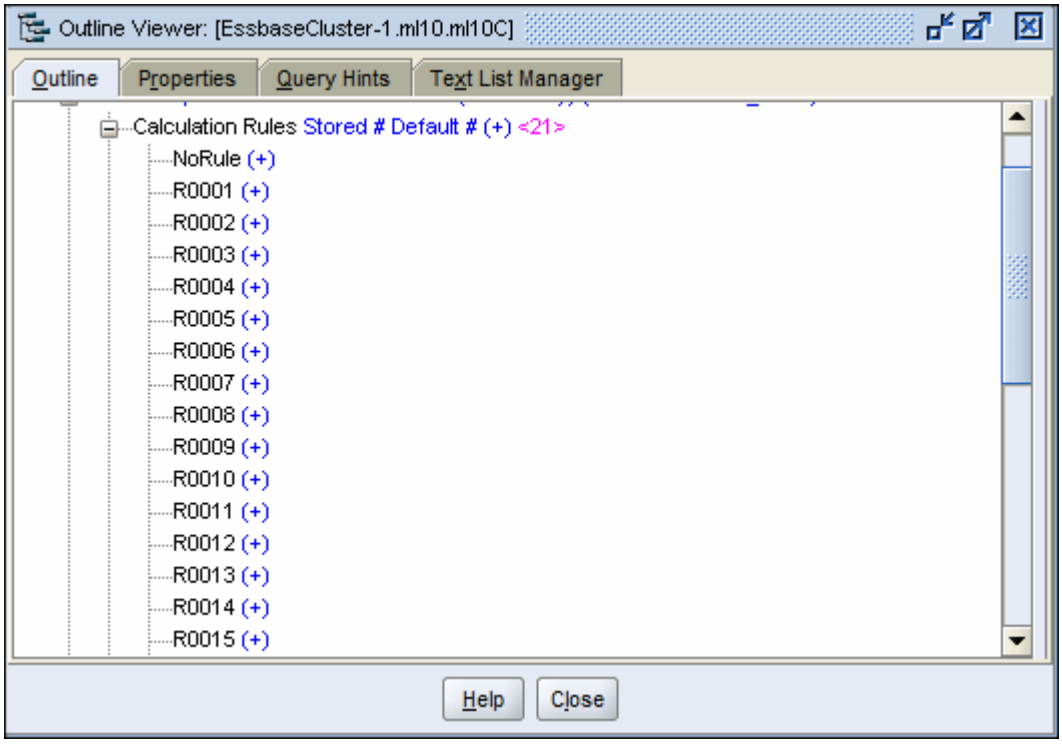
> **Note:**
>
> The Management Ledger Rule and Balance dimensions are system dimensions that are seeded and should not be edited in any way, even if some part of the system allows it (for example, Update Dimensions in the Profitability Applications Console). These dimensions are reserved for system use.

Management Ledger Rule Dimension

The Rule dimension contains definitions of calculation rules for Management Ledger applications.

Figure 1 shows the outline of the Rule dimension in the Oracle Essbase console. This is an illustration of the Calculation Rules members.

Figure 5-2 The Management Ledger Rule Dimension Outline



> **Note:**
>
> For Releases 11.2.15 or later, ensure that you use the Essbase 21c Web Console Outline Screen.

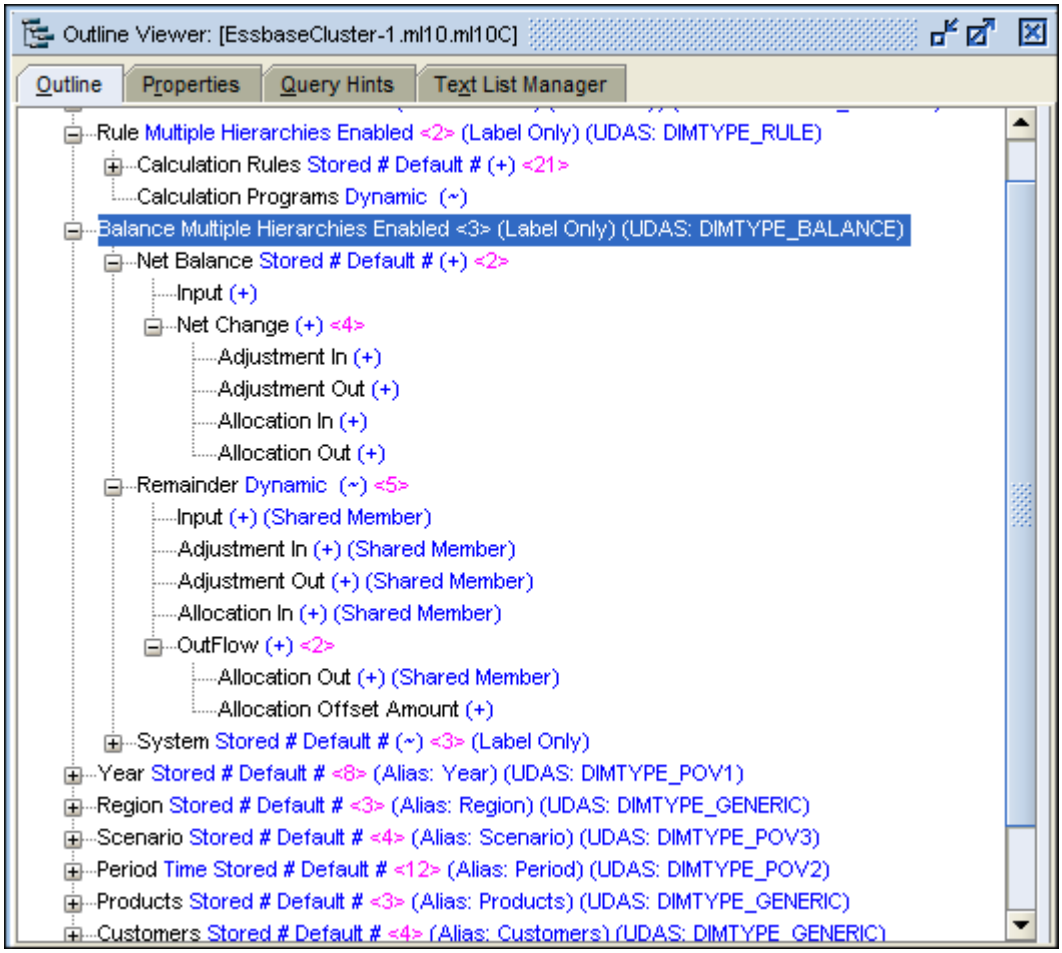
Users can delete and add rules to Calculation Rules, but the only rule they can apply is NoRule. All the others are reserved for system use.

Figure 1 shows an additional member, Calculation Programs. This member is controlled by the system and is not editable by users.

Management Ledger Balance Dimension

Figure 1 shows the outline of the Balance dimension in the Oracle Essbase console.

Figure 5-3 The Management Ledger Balance Dimension Outline



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.

Management Ledger Business Dimensions

Business dimensions describe the business-specific objects within the model, such as products, customers, regions, and employees.

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 Oracle Essbase outlines are deployed, the business dimensions are created in the Oracle Hyperion 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.
>
> The Management Ledger Rule and Balance dimensions are system dimensions that are seeded and should not be edited in any way, even if some part of the system allows it (for example, Update Dimensions in the Profitability Applications Console). These dimensions are reserved for system use.

Business Dimension Requirements

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.

> **Note:**
>
> In Management Ledger business dimensions, do not make shared members that reference non-level-0 base members. Doing so will cause the Copy POV and the calculation operations to fail because sharing with such references creates situations where Profitability and Cost Management needs to write to non-level-0 members, which is not supported in Essbase for ASO cubes.

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.

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 Essbase Naming Conventions.

There are two Attribute dimension types:

- 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 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.

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.

When working with Management Ledger alias dimensions, keep in mind the following points:

- Duplicate member names or aliases are not allowed within the same dimension.
- When when importing a dimension file into a Management Ledger application, if you define an alias table in the header, then you must define alias values for all of the members in the dimension. The alias value does not necessarily have to be different from the original member name, but you do need to specify a value for each alias table on each member row in the file.

Management Ledger Dimension Sort Order

The Dimension Sort Order property controls the order of dimensions in the Oracle Essbase outline that is generated by Oracle Hyperion 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 the dimension load file. For instructions, see Setting the Dimension Sort Order.

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 Oracle Hyperion Profitability and Cost Management and Oracle 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 a Dimension Sort Order property in the dimension load files.

The dimension sort order restrictions must be met; otherwise, validation of the model fails. For a complete list of restrictions, see Dimension Sort Order Recommendations.

To construct a load file that includes the dimension sort order:

1. In an external program or ETL process, construct a load file with columns similar to these. Make sure to include a column for the Dimension Sort Order.

   Enter a sort order for the root member of the dimension. In this example, *Accounts* has sort order of *9*.

Regular	Parent	Storage Type	Hierarchy Type	Comment	ASO Data Storage	BSO Data Storage	Two Pass Calculation	ASO Dimension Formula	Consolidation Type	UDA	Dimension Solve Order	Dimension Sort Order	Member Solve Order	Alias:English	Alias:Default
Accounts		SPARSE	STORED		LABELONLY	LABELONLY	N				0	9		ALIAS:English	ALIAS:Default
AllAccounts		SPARSE	STORED		StoreData	StoreData	N		+				0	AllAccounts	AllAccounts
"NoAccour	AllAccoun	SPARSE	STORED		StoreData	StoreData	N		+				0	NoAccounts	NoAccounts
"Cost of G	AllAccoun	SPARSE	STORED		StoreData	StoreData	N		+				0	Cost of Goods	Cost of Goods
"Material	Cost of Gc	SPARSE	STORED		StoreData	StoreData	N		+				0	Materials	Materials
"MAT51(	Materials	SPARSE	STORED		StoreData	StoreData	N		+				0	Frames	Frames
"MAT52(	Materials	SPARSE	STORED		StoreData	StoreData	N		+				0	Wheels	Wheels
"MAT53(	Materials	SPARSE	STORED		StoreData	StoreData	N		+				0	Gearsets	Gearsets
"MAT54(	Materials	SPARSE	STORED		StoreData	StoreData	N		+				0	Seats	Seats
"MAT55(	Materials	SPARSE	STORED		StoreData	StoreData	N		+				0	Controls	Controls
"MAT56(	Materials	SPARSE	STORED		StoreData	StoreData	N		+				0	Tubing	Tubing
"Equipme	Cost of Gc	SPARSE	STORED		StoreData	StoreData	N		+				0	Equipment De	Equipment Dep
"DEP610	Equipmen	SPARSE	STORED		StoreData	StoreData	N		+				0	Bar Bender De	Bar Bender Dep
"Personne	AllAccoun	SPARSE	STORED		StoreData	StoreData	N		+				0	Personnel Exp	Personnel Expe
"PER2100	Personnel	SPARSE	STORED		StoreData	StoreData	N		+				0	Regular Salary	Regular Salary
"PER2200	Personnel	SPARSE	STORED		StoreData	StoreData	N		+				0	Regular Wages	Regular Wages
"PER2300	Personnel	SPARSE	STORED		StoreData	StoreData	N		+				0	Overtime	Overtime
"PER2400	Personnel	SPARSE	STORED		StoreData	StoreData	N		+				0	Vacation	Vacation

2. In the Profitability Applications Console, select Actions, and then **Update Dimensions** to load the file with the dimension sort order defined.

6

Performing Other Application Management Tasks

Related Topics

- Managing Management Ledger Profitability Applications
- Validating and Deploying Management Ledger Applications Using the Wizard
  After creating a Oracle Hyperion Profitability and Cost Management application, the application must be validated and deployed to Profitability and Cost Management.

Managing Management Ledger Profitability Applications

Related Topics

- Working with Management Ledger Applications and Dimensions
- Profitability and Cost Management Dimension and Member Properties
- Setting Hierarchy Type Property

Working with Management Ledger Applications and Dimensions

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

Profitability and Cost Management Dimension and Member Properties

The properties for Oracle Hyperion Profitability and Cost Management dimensions and members are displayed in alphabetical order in the Management Ledger and Cost Management Dimension and Member Properties table 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.

> **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 Oracle Hyperion Enterprise Performance Management 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 Oracle Hyperion Profitability and Cost Management, and then Oracle Essbase.

Validating and Deploying Management Ledger Applications Using the Wizard

After creating a Oracle Hyperion Profitability and Cost Management application, 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.

Part IV
Working with Detailed Profitability

Related Topics

- Understanding 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
- Calculating Detailed Profitability Models
- Importing Detailed Profitability Staging Tables
- Exporting Model Definition Data for Detailed Profitability

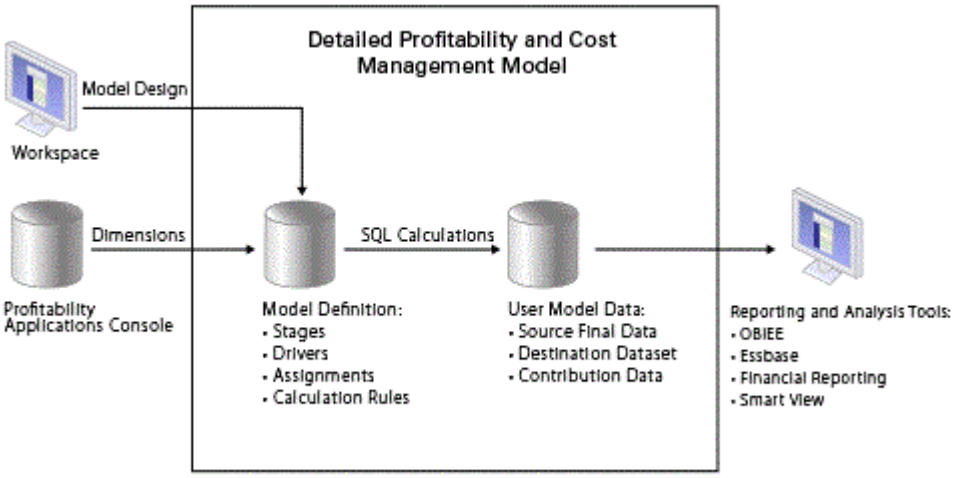
7

Understanding Detailed Profitability Product Architecture

Accessed through Oracle Hyperion Enterprise Performance Management 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.

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

Figure 7-1 Detailed Profitability Architecture



Application administrators create the Profitability and Cost Management dimensions using the Profitability Applications Console. (See Creating Applications Using the Profitability Applications Console). 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 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. See Product Schema.
- A schema, referred to as the Model Data Schema, for storing existing data and the results of the model calculations. See Model Data Schema for Detailed Profitability.

8

Working with Detailed Profitability Databases

For Detailed Profitability applications, the dimensional data and model definition are stored in the same relational database schema that is used to store dimensional data and model definitions for Management Ledger 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 the application from the Profitability Applications Console. Model definitions are stored in this schema as you build the model.

For Detailed Profitability applications, the business data upon which allocations are performed is also stored in the relational database (not in Oracle Essbase as is the case for Management Ledger 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, the dimensions and members are mapped to columns and registered to the application tables in Oracle Hyperion Profitability and Cost Management. For information on Detailed Profitability dimensions, see Working with Detailed Profitability and Cost Management Dimensions and Metadata.

Table 8-1 Detailed Profitability and Cost Management Database Schemas

Schema Type	Information Type Stored
Model Data Schema	• Dimension Hierarchies • Cost Data • Revenue Data • Driver Data • Profit Object Data • Currency Rates
Product Schema	• Dimensions • Stage Definitions • POV Definitions • Driver Definitions • Driver Selections • Assignment Rules • Assignment Rules Selections • Model Preferences • Table Registrations • Calculation Rules

9
Product Schema

Detailed Profitability applications shares the same Product Schema that holds the Oracle Hyperion 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 Microsoft SQL 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 Profitability and Cost Management product schema. It will change the default schema for that user, which must remain set as the Profitability and Cost Management 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.

10

Model Data Schema for Detailed Profitability

Related Topics

- About Model Data Schema for Detailed Profitability
  The Model Data Schema can be a pre-existing schema or one created by your Database Administrator to support your Detailed Profitability applications.
- Model Data Schema Table Grants to the Product Schema
  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.
- 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 some requirements.
- Registering Database Views
  If you made the appropriate grants to the Product Schema, you may also register database views from the Model Data Schema.

About Model Data Schema for Detailed Profitability

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. Oracle Hyperion 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

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 10-1 Privileges Granted for Each Table Type

Table Type	Privileges
Source Stage Table either Horizontal or Vertical	`GRANT SELECT ON <table_name or view_name> TO <hpm_product_schema> WITH GRANT OPTION`
Destination Stage Table	`GRANT SELECT, UPDATE ON <table_name or view_name> TO <hpm_product_schema>`
All types of Lookup Tables	`GRANT SELECT ON <table_name or view_name> TO <hpm_product_schema>`

> **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, 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 the Profitability Applications Console. For information on Detailed Profitability dimensions, see 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 some 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, Oracle Hyperion 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 the destination stage table and lookup tables with 4 or more decimal places to avoid rounding errors.

Caution:

Before you import data or artifacts, you must create a backup of the Model Data Schema. See the *Oracle Enterprise Performance Management System Backup and Recovery Guide*

11

Working with Detailed Profitability and Cost Management Dimensions and Metadata

Related Topics

- About Detailed Profitability Dimensions and Metadata
  Detailed Profitability enables you to use the existing relational database as the starting point for your Detailed Profitability application.
- Detailed Profitability Dimension Types
  For Detailed Oracle Hyperion Profitability and Cost Management, there are several types of available dimensions.
- Alias Dimensions
  Aliases are alternate names, descriptions, languages, or other items that help to define dimensions.
- Business Dimensions
  Business dimensions describe the business-specific objects within each stage in the model, such as products, customers, regions, and employees.
- 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.
- 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.
- Dimension Sort Order and Density Settings
  The Dimension Sort Order property controls the order of dimensions in the Oracle Essbase outline that is generated by Oracle Hyperion Profitability and Cost Management.
- Setting the Dimension Sort Order
  The processing order for every dimension in the model must be set at the dimension level using a Dimension Sort Order property in the dimension load files.

About Detailed Profitability Dimensions and Metadata

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

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

Profitability and Cost Management uses the dimensions and members created in the Profitability Applications Console 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): A reserved dimension that contains the allocation members require to support stage balancing and other verification activities. Provides the structure for the model and scenarios This dimension does not contain driver measures.
- At least one POV dimension (required): Time periods, scenarios and versions
- At least one Business Dimension (required): Reflects the business-specific elements of the model, such as departments, General Ledger accounts, activities, customers, or products
- 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.
- 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 Essbase Naming Conventions.

> **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 Creating Applications Using the Profitability Applications Console.

Detailed Profitability Dimension Types

For Detailed Oracle Hyperion Profitability and Cost Management, there are several types of available dimensions.

- 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 the 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.

> **Note:**
>
> When defining dimensional outlines, there are restricted characters that may not be used for naming. Oracle strongly suggests that you review the Oracle 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.

Aliases may be cloned if a particular dimension is cloned in Oracle 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. 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, and employees.

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 Oracle Essbase outlines are deployed, the business dimensions are created in the Oracle Hyperion 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, 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 (~)`.

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.

A Version dimension is also available, and is used to create another instance of the 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 Essbase Naming Conventions.

There are two Attribute dimension types:

- 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 Essbase Naming Conventions.

Dimension Sort Order and Density Settings

The Dimension Sort Order property controls the order of dimensions in the Oracle Essbase outline that is generated by Oracle Hyperion 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 the dimension load files. For instructions, see Setting the Dimension Sort Order.

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 Oracle Hyperion Profitability and Cost Management and Oracle 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 a Dimension Sort Order property in the dimension load files.

The dimension sort order restrictions must be met; otherwise, validation of the model fails. For a complete list of restrictions, see Dimension Sort Order Recommendations.

To construct a load file that includes the dimension sort order:

1. In an external program or ETL process, construct a load file with columns similar to these. Make sure to include a column for the Dimension Sort Order.

   Enter a sort order for the root member of the dimension. In this example, *Accounts* has sort order of *9*.

Regular	Parent	Storage Type	Hierarchy Type	Comment	ASO Data Storage	BSO Data Storage	Two Pass Calculation	ASO Dimension Formula	Consolidation Type	UDA	Dimension Solve Order	Dimension Sort Order	Member Solve Order	Alias:English	Alias:Default
Accounts		SPARSE	STORED		LABELONLY	LABELONLY	N				0	9		ALIAS:English	ALIAS:Default
AllAccounts		SPARSE	STORED		StoreData	StoreData	N		+				0	AllAccounts	AllAccounts
"NoAccour	AllAccoun	SPARSE	STORED		StoreData	StoreData	N		+				0	NoAccounts	NoAccounts
"Cost of G	AllAccoun	SPARSE	STORED		StoreData	StoreData	N		+				0	Cost of Goods	Cost of Goods
"Material	Cost of Gc	SPARSE	STORED		StoreData	StoreData	N		+				0	Materials	Materials
"MAT51I	Materials	SPARSE	STORED		StoreData	StoreData	N		+				0	Frames	Frames
"MAT52I	Materials	SPARSE	STORED		StoreData	StoreData	N		+				0	Wheels	Wheels
"MAT53I	Materials	SPARSE	STORED		StoreData	StoreData	N		+				0	Gearsets	Gearsets
"MAT54I	Materials	SPARSE	STORED		StoreData	StoreData	N		+				0	Seats	Seats
"MAT55I	Materials	SPARSE	STORED		StoreData	StoreData	N		+				0	Controls	Controls
"MAT56I	Materials	SPARSE	STORED		StoreData	StoreData	N		+				0	Tubing	Tubing
"Equipme	Cost of Gc	SPARSE	STORED		StoreData	StoreData	N		+				0	Equipment De	Equipment Dep
"DEP610	Equipmen	SPARSE	STORED		StoreData	StoreData	N		+				0	Bar Bender De	Bar Bender Dep
"Personne	AllAccoun	SPARSE	STORED		StoreData	StoreData	N		+				0	Personnel Exp	Personnel Expe
"PER2100	Personnel	SPARSE	STORED		StoreData	StoreData	N		+				0	Regular Salary	Regular Salary
"PER2200	Personnel	SPARSE	STORED		StoreData	StoreData	N		+				0	Regular Wages	Regular Wages
"PER2300	Personnel	SPARSE	STORED		StoreData	StoreData	N		+				0	Overtime	Overtime
"PER2400	Personnel	SPARSE	STORED		StoreData	StoreData	N		+				0	Vacation	Vacation

2. In the Profitability Applications Console, select Actions, and then **Update Dimensions** to load the file with the dimension sort order defined.

12

Calculating Detailed Profitability Models

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 Oracle Hyperion Profitability and Cost Management. You must redeploy these tables each time you redeploy a Detailed Profitability application. 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 mModel 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
- Creating Custom Scripts

HPM_SQL_SCRIPT

This table stores the 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 the script directly into the HPM_SQL_SCRIPT. See Creating Custom Scripts.

Table 12-1 HPM_SQL_SCRIPT

Column	Data Type	Nullable	Description
NAME	Varchar2 (80)	No	Name of the custom script
APPLICATION_NAME	Varchar2 (8)	No	Name of the application for which this custom script has been created
MODEL_POV_NAME	Varchar2 (80)	Yes	The selected POV is displayed in the format, `"Year":"2012", "Period":"January", "Scenario":"Actual".` **Note:** 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.
SCRIPT_TYPE	Varchar2 (80)	Yes	Select the appropriate script type: • PRE (displays Pre-Script) • POST (displays Post-Script) • EITHER of (displays Either Pre-Script or Post-Script) • If no script type is specified, EITHER is assumed.
DESCRIPTION	Varchar2 (255)	Yes	Enter a description of the purpose or contents of the script
SCRIPT	LONG	No	Enter the custom script here. See Creating Custom Scripts 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. 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 the 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.
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 Oracle Hyperion 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
- Other Process Type

> **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 Oracle Hyperion 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.

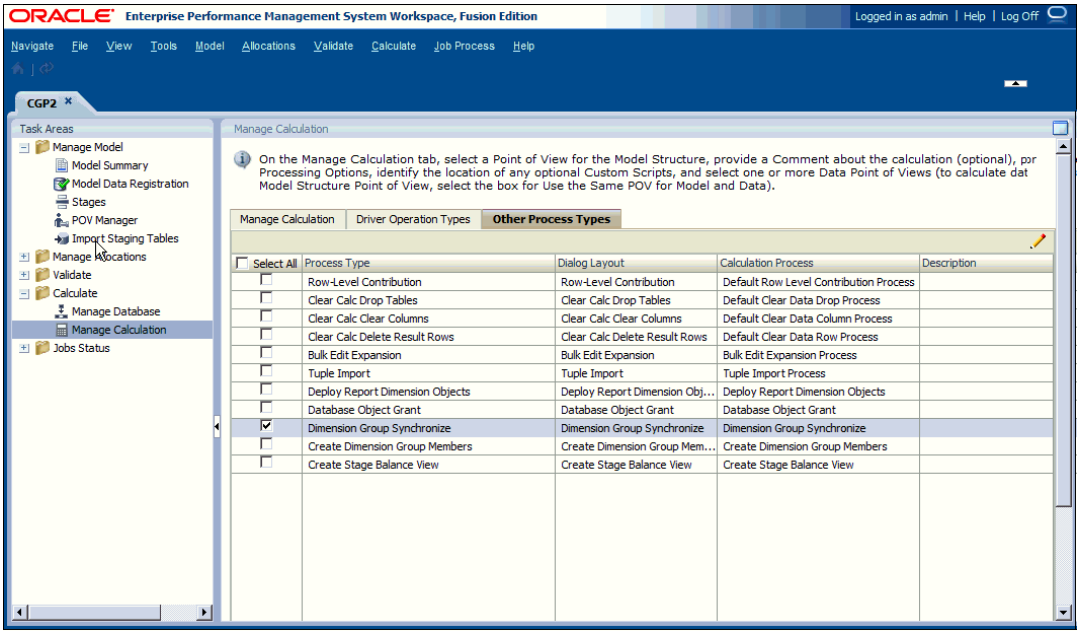
To change your Oracle Database version:

1. From Oracle Hyperion Enterprise Performance Management 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.

13

Importing Detailed Profitability Staging Tables

Related Topics

- About Importing Detailed Profitability Staging Tables
  To import model data from relational databases into Oracle Hyperion 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.
- Creating Import Database Tables for Detailed Profitability
  Staging database scripts are available for Microsoft SQL Server and Oracle Database
- HPM_STGD_POV
  The HPM_STGD_POV table stores the states of each combination of dimension members included in a point of view (POV).
- HPM_STGD_DRIVER
  The HPM_STGD_DRIVER table provides details about the driver, including driver type, display order and formula.
- 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.
- HPM_STGD_DRIVER_EXCEP
  The HPM_STGD_DRIVER_EXCEP table provides details about the POV and source stage intersection for the selected driver exceptions.
- HPM_STGD_ASGN_RULE_SEL
  The HPM_STGD_ASGN_RULE_SEL table provides details about the assignment rule selections.
- HPM_STGD_CALCRULE_SNGLSRC
  The HPM_STGD_CALCRULE_SNGLSRC table provides details about creating or updating a single source calculation rule.
- HPM_STGD_CALCRULE_CALCMSRS
  The HPM_STGD_CALCRULE_CALCMSRS table provides details about creating or updating a calculated measures rule.
- HPM_STGD_CALCRULE_MULTISRC
  The HPM_STGD_CALCRULE_MULTISRC table provides details about creating or updating a multi source calculation rule.

About Importing Detailed Profitability Staging Tables

To import model data from relational databases into Oracle Hyperion 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 Profitabilit). 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:

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 13-1 HPM_STGD_POV

Field	SQL Server Data Type	Oracle Data Type	Required	Description
id	identity	integer(38,0)	Yes	Unique record ID
pov_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	Yes	Name of the first dimension member in the selected POV
pov_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)		Name of the second dimension member in the selected POV
pov_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)		Name of the third dimension member in the selected POV
pov_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)		Name of the fourth dimension member in the selected POV
pov_state	nvarchar (80)	varchar2(80 CHAR)	Yes	The current status of the POV: • Draft • Published • Archived

Table 13-1 (Cont.) HPM_STGD_POV

Field	SQL Server Data Type	Oracle Data Type	Required	Description
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)	Populated by the import program upon update	The ID of the user who initiated the last import
created_timestamp	timestamp (6)	timestamp (6)		The date and time of the last import
modified_userid	nvarchar (32)	varchar2(32 CHAR)	Populated by the import program upon update	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_STGD_DRIVER

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

Table 13-2 HPM_STGD_DRIVER

Field	SQL Server Data Type	Oracle Data Type	Required	Description
id	identity	integer(38,0)	Yes	Unique record ID
name	nvarchar (80)	varchar2(80 CHAR)	Yes	Name of the selected driver
description	nvarchar (255)	varchar2(255 CHAR)		Description of the purpose for the selected driver
formula	nvarchar (500)	varchar2(500 CHAR)	Yes	Formula created for the driver by a user. This formula must be created using SQL syntax, and mathematically correct operations.

Table 13-2 (Cont.) HPM_STGD_DRIVER

Field	SQL Server Data Type	Oracle Data Type	Required	Description
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. 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.
dst_measure_member_name	nvarchar (80)	varchar2(80 CHAR)	Yes	Name of the measure on the Destination table
tdv_measure_member_name	nvarchar (80)	varchar2(80 CHAR)		Measure that is used as the denominator in the allocation formula:P `Driver Value/ TotalDriverValue` (DV/TDV)
operation_type	nvarchar (80)	varchar2(80 CHAR)	Yes	The operation type selected for the driver: • Rate-based Allocation • Ratio-based Allocation • Calculated Measures
last_upload_date	timestamp (6)	timestamp (6)		The date and time of the last import

Table 13-2 (Cont.) HPM_STGD_DRIVER

Field	SQL Server Data Type	Oracle Data Type	Required	Description
exception	nvarchar (255)	varchar2(255 CHAR)		Message detailing errors that occurred during the import of this table.
created_userid	nvarchar (32)	varchar2(32 CHAR)	Populated by the import program upon update	The ID of the user who initiated the last import
created_timestamp	timestamp (6)	timestamp (6)		The date and time of the last import
modified_userid	nvarchar (32)	varchar2(32 CHAR)	Populated by the import program upon update	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_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 13-3 HPM_STGD_DRIVER_SEL

Field	SQL Server Data Type	Oracle Data Type	Required	Description
id	identity	integer(38,0)	Yes	Unique record ID
pov_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	Yes	Name of the first dimension member in the selected POV
pov_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)		Name of the second dimension member in the selected POV, if applicable
pov_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)		Name of the third dimension member in the selected POV, if applicable
pov_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)		Name of the fourth dimension member in the selected POV, if applicable
driver_dim_member_name	nvarchar (80)	varchar2(80 CHAR)	Yes	Name of the selected driver dimension member
driver_name	nvarchar (80)	varchar2(80 CHAR)	Yes	Name of the selected driver
last_upload_date	timestamp (6)	timestamp (6)		The date and time of the last import

Table 13-3 (Cont.) HPM_STGD_DRIVER_SEL

Field	SQL Server Data Type	Oracle Data Type	Required	Description
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)	Populated by the import program upon update	The ID of the user who initiated the last import
created_timestamp	timestamp (6)	timestamp (6)		The date and time of the last import
modified_userid	nvarchar (32)	varchar2(32 CHAR)	Populated by the import program upon update	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_STGD_DRIVER_EXCEP

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

Table 13-4 HPM_STGD_DRIVER_EXCEP

Field	SQL Server Data Type	Oracle Data Type	Required	Description
id	identity	integer(38,0)	Yes	Unique record ID
pov_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	Yes	Name of the first dimension member in the selected POV
pov_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)		Name of the second dimension member in the selected POV, if applicable
pov_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)		Name of the third dimension member in the selected POV, if applicable
pov_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)		Name of the fourth dimension member in the selected POV, if applicable
src_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	Yes	Name of the first dimension member in the source stage
src_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)		Name of the second dimension member in the source stage, if applicable

Table 13-4 (Cont.) HPM_STGD_DRIVER_EXCEP

Field	SQL Server Data Type	Oracle Data Type	Required	Description
src_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)		Name of the third dimension member in the source stage, if applicable
src_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)		Name of the fourth dimension member in the source stage, if applicable
src_dim5_member_name	nvarchar (80)	varchar2(80 CHAR)		Name of the fifth dimension member in the source stage, if applicable
driver_name	nvarchar (80)	varchar2(80 CHAR)	Yes	Name of the selected driver
last_upload_date	timestamp (6)	timestamp (6)		The date and time of the last import
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)	Populated by the import program upon update	The ID of the user who initiated the last import
created_timestamp	timestamp (6)	timestamp (6)		The date and time of the last import
modified_userid	nvarchar (32)	varchar2(32 CHAR)	Populated by the import program upon update	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_STGD_ASGN_RULE_SEL

The HPM_STGD_ASGN_RULE_SEL table provides details about the assignment rule selections.

Table 13-5 HPM_STGD_ASGN_RULE_SEL

Field	SQL Server Data Type	Oracle Data Type	Required	Description
id	identity	integer(38,0)	Yes	Unique record ID
pov_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	Yes	Name of the first dimension member in the selected POV
pov_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)		Name of the second dimension member in the selected POV

Table 13-5 (Cont.) HPM_STGD_ASGN_RULE_SEL

Field	SQL Server Data Type	Oracle Data Type	Required	Description
pov_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)		Name of the third dimension member in the selected POV
pov_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)		Name of the fourth dimension member in the selected POV
src_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	Yes	Name of the first dimension member in the source stage
src_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)		Name of the second dimension member in the source stage, if applicable
src_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)		Name of the third dimension member in the source stage, if applicable
src_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)		Name of the fourth dimension member in the source stage, if applicable
src_dim5_member_name	nvarchar (80)	varchar2(80 CHAR)		Name of the fifth dimension member in the source stage, if applicable
rule_name	nvarchar (80)	varchar2(80 CHAR)	Yes	Name of the rule controlling the selected assignment. The rule must be present in the target database.
last_upload_date	timestamp (6)	timestamp (6)		The date and time of the last import
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)	Populated by the import program upon update	The ID of the user who initiated the last import
created_timestamp	timestamp (6)	timestamp (6)		The date and time of the last import
modified_userid	nvarchar (32)	varchar2(32 CHAR)	Populated by the import program upon update	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_STGD_CALCRULE_SNGLSRC

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

Table 13-6 HPM_STGD_CALCRULE_SNGLSRC

Field	SQL Server Data Type	Oracle Data Type	Required	Description
id	identity	NUMBER(38,0)	Yes	Unique identifier for each row. This can come from any sequence generator.
name	nvarchar (80)	VARCHAR2(80 CHAR)	Yes	Name of the calculation rule.
description	nvarchar (255)	varchar2(255 CHAR)		Long description of the calculation rule.
pov_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	Yes	Name of the first dimension member in the selected POV
pov_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)		Name of the second dimension member in the selected POV
pov_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)		Name of the third dimension member in the selected POV
pov_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)		Name of the fourth dimension member in the selected POV
calculation_sequence	identity	NUMBER(4,0)	Yes	Numeric value designating the relative order in which the calculation rule will be processed during calculation. This must be > 0 or you will get a check constraint error during insert.
tag	nvarchar (255)	VARCHAR2(255)		Specify search tags for the calculation rule in this column.
enabled_flag	nvarchar (1)	VARCHAR2(1 CHAR)	Yes	
first_sequence_priority	identity	NUMBER(38,0)	Yes	Designates the first driver sequence priority in the range of drivers to be executed for this rule. This must be > 0 or you will get a check constraint error during insert.
last_sequence_priority	identity	NUMBER(38,0)	Yes	Designates the last driver sequence priority in the range of drivers to be executed for this rule. This must be equal to or greater than `first_sequence_priority` or you will get a check constraint error during insert.
all_drivers_flag	nvarchar (1)	VARCHAR2(1 CHAR)	Yes	When Y ignore values in the First Sequence and Last Sequence priority fields during calculation.
last_upload_date	timestamp	timestamp		This column will be populated by the import showing the most recent date that the import was performed.
import_exception	nvarchar (255)	varchar2(255 CHAR)	.	Contains the error code indicating if there was a problem with the import of the row. Null means the import was successful.

Table 13-6 (Cont.) HPM_STGD_CALCRULE_SNGLSRC

Field	SQL Server Data Type	Oracle Data Type	Required	Description
created_userid	nvarchar (32)	varchar2(32 CHAR)		Identifies the user who originally created the row in this table. This is not used by Profitability and Cost Management.
created_timestamp	timestamp (6)	timestamp		Identifies the date/time on which the row was originally created. This is not used by Profitability and Cost Management.
modified_userid	nvarchar (32)	varchar2(32 CHAR)		Identifies the user who last modified the row in this table. This is not used by Profitability and Cost Management.
modified_timestamp	timestamp (6)	timestamp		Identifies the date/time on which the row was last changed. This is not used by Profitability and Cost Management.

HPM_STGD_CALCRULE_CALCMSRS

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

Table 13-7 HPM_STGD_CALCRULE_CALCMSRS

Field	SQL Server Data Type	Oracle Data Type	Required	Description
id	identity	number(38,0)	Yes	Unique identifier for each row. This can come from any sequence generator.
name	nvarchar (80)	varchar2(80 CHAR)	Yes	Name of the calculation rule.
description	nvarchar (255)	varchar2(255 CHAR)		Long description of the calculation rule.
pov_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	Yes	Name of the first dimension member in the selected POV
pov_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)		Name of the second dimension member in the selected POV
pov_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)		Name of the third dimension member in the selected POV
pov_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)		Name of the fourth dimension member in the selected POV

Table 13-7 (Cont.) HPM_STGD_CALCRULE_CALCMSRS

Field	SQL Server Data Type	Oracle Data Type	Required	Description
calculation_sequence	number	number(4,0)	Yes	Numeric value designating the relative order in which the calculation rule will be processed during calculation. This must be > 0 or you will get a check constraint error during insert.
tag	nvarchar (255)	varchar2(255)		Specify search tags for the calculation rule in this column.
enabled_flag	nvarchar (1)	varchar2(1 CHAR)	Yes	
driver_name1	nvarchar (1)	varchar2(1 CHAR)	Yes	Driver name to be associated with this calculation rule.
driver_name2	nvarchar (80)	varchar2(80 CHAR)		Driver name to be associated with this calculation rule.
driver_name3	nvarchar (80)	varchar2(80 CHAR)		Driver name to be associated with this calculation rule.
driver_name4	nvarchar (80)	varchar2(80 CHAR)		Driver name to be associated with this calculation rule.
driver_name5	nvarchar (80)	varchar2(80 CHAR)		Driver name to be associated with this calculation rule.
dest_assign_rule_name1	nvarchar (80)	varchar2(80 CHAR)	Yes	Destination Assignment Rule to be associated with this calculation rule.
dest_assign_rule_name2	nvarchar (80)	varchar2(80 CHAR)		Destination Assignment Rule to be associated with this calculation rule.
dest_assign_rule_name3	nvarchar (80)	varchar2(80 CHAR)		Destination Assignment Rule to be associated with this calculation rule.
dest_assign_rule_name4	nvarchar (80)	varchar2(80 CHAR)		Destination Assignment Rule to be associated with this calculation rule.
dest_assign_rule_name5	nvarchar (80)	varchar2(80 CHAR)		Destination Assignment Rule to be associated with this calculation rule.

Table 13-7 (Cont.) HPM_STGD_CALCRULE_CALCMSRS

Field	SQL Server Data Type	Oracle Data Type	Required	Description
last_upload_date	timestamp (6)	timestamp		This column will be populated by the import showing the most recent date that the import was performed.
import_exception	nvarchar (255)	varchar2(255 CHAR)		Contains the error code indicating if there was a problem with the import of the row. Null means the import was successful.
created_userid	nvarchar (32)	varchar2(32 CHAR)		Identifies the user who originally created the row in this table. This is not used by Profitability and Cost Management..
created_timestamp	timestamp (6)	timestamp		Identifies the date/time on which the row was originally created. This is not used by Profitability and Cost Management.
modified_userid	nvarchar (32)	varchar2(32 CHAR)		Identifies the user who originally created the row in this table. This is not used by Profitability and Cost Management.
modified_timestamp	timestamp (6)	timestamp		Identifies the date/time on which the row was originally created. This is not used by Profitability and Cost Management.

HPM_STGD_CALCRULE_MULTISRC

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

Table 13-8 HPM_STGD_CALCRULE_CALCMSRS

Field	SQL Server Data Type	Oracle Data Type	Required	Description
id	identity	number(38,0)	Yes	Unique identifier for each row. This can come from any sequence generator.
name	nvarchar (80)	varchar2(80 CHAR)	Yes	Name of the calculation rule.

Table 13-8 (Cont.) HPM_STGD_CALCRULE_CALCMSRS

Field	SQL Server Data Type	Oracle Data Type	Required	Description
description	nvarchar (255)	varchar2(255 CHAR)		Long description of the calculation rule.
pov_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	Yes	Name of the first dimension member in the selected POV
pov_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)		Name of the second dimension member in the selected POV
pov_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)		Name of the third dimension member in the selected POV
pov_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)		Name of the fourth dimension member in the selected POV
calculation_sequence	number	number(4,0)	Yes	Numeric value designating the relative order in which the calculation rule will be processed during calculation. This must be > 0 or you will get a check constraint error during insert.
tag	nvarchar (255)	varchar2(255)		Specify search tags for the calculation rule in this column.
enabled_flag	nvarchar (1)	varchar2(1 CHAR)	Yes	
driver_name	nvarchar (80)	varchar2(80 CHAR)	Yes	Driver name to be associated with this calculation rule.
src_assign_rule_name1	nvarchar (80)	varchar2(80 CHAR)	Yes	Source assignment rule to be associated with this calculation rule.
src_assign_rule_name2	nvarchar (80)	varchar2(80 CHAR)		Source assignment rule to be associated with this calculation rule.
src_assign_rule_name3	nvarchar (80)	varchar2(80 CHAR)		Source assignment rule to be associated with this calculation rule.
src_assign_rule_name4	nvarchar (80)	varchar2(80 CHAR)		Source assignment rule to be associated with this calculation rule.
src_assign_rule_name5	nvarchar (80)	varchar2(80 CHAR)		Source assignment rule to be associated with this calculation rule.
dest_assign_rule_name1	nvarchar (80)	varchar2(80 CHAR)	Yes	Destination assignment rule to be associated with this calculation rule.

Table 13-8 (Cont.) HPM_STGD_CALCRULE_CALCMSRS

Field	SQL Server Data Type	Oracle Data Type	Required	Description
dest_assign_rule_name2	nvarchar (80)	varchar2(80 CHAR)		Destination Assignment Rule to be associated with this calculation rule.
dest_assign_rule_name3	nvarchar (80)	varchar2(80 CHAR)		Destination assignment rule to be associated with this calculation rule.
dest_assign_rule_name4	nvarchar (80)	varchar2(80 CHAR)		Destination assignment rule to be associated with this calculation rule.
dest_assign_rule_name5	nvarchar (80)	varchar2(80 CHAR)		Destination assignment rule to be associated with this calculation rule.
last_upload_date	timestamp (6)	timestamp		This column will be populated by the import showing the most recent date that the import was performed.
import_exception	nvarchar (255)	VARCHAR2(255 CHAR)		Contains the error code indicating if there was a problem with the import of the row. Null means the import was successful.
created_userid	nvarchar (32)	varchar2(32 CHAR)		Identifies the user who originally created the row in this table. This is not used by Profitability and Cost Management.
created_timestamp	timestamp (6)	timestamp		Identifies the date/time on which the row was originally created. This is not used by Profitability and Cost Management.
modified_userid	nvarchar (32)	varchar2(32 CHAR)		Identifies the user who originally created the row in this table. This is not used by Profitability and Cost Management.
modified_timestamp	timestamp (6)	timestamp		Identifies the date/time on which the row was originally created. This is not used by Profitability and Cost Management.

14

Exporting Model Definition Data for Detailed Profitability

Related Topics

- About Exporting Model Definition Data for Detailed Profitability
  After the model has been created, you can query the database to display the model artifacts as output in a database view.
- 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.
- HPM_EXPD_POV
  The HPM_EXPD_POV view retrieves the states of each combination of dimension members included in a point of view (POV).
- HPM_EXPD_DRIVER
  The HPM_EXPD_DRIVER view retrieves details about the driver, including driver type, display order and formula.
- HPM_EXPD_DRIVER_SEL
  The HPM_EXPD_DRIVER_SEL view retrieves details about the POV, and the driver selections.
- HPM_EXPD_DRIVER_EXCEP
  The HPM_EXPD_DRIVER_EXCEP view retrieves details about the POV and source stage for the selected driver exceptions.
- HPM_EXPD_ASGN_RUL_SEL
  The HPM_EXPD_ASGN_RULE_SEL view retrieves details about the assignment rule selections.
- HPM_EXPD_CALCRULE_SNGLSRC
  The HPM_EXPD_CALCRULE_SNGLSRC view retrieves details about the single source calculation rule.
- HPM_EXPD_CALCRULE_CALCMSRS
  The HPM_EXPD_CALCRULE_CALCMSRS view retrieves details about the calculated measures rule.
- HPM_EXPD_CALCRULE_MULTISRC
  The HPM_EXPD_MULTISRC view retrieves details about the multi source calculation rule.
- 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.

About Exporting Model Definition Data for Detailed Profitability

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 14-1 HPM_EXPD_STAGE

Field	SQL Server Data Type	Oracle Data Type	Description
application_name	nvarchar (80)	varchar2(80 CHAR)	Name of the selected application
id	identity	integer(38,0)	Stage ID that is used in the Oracle Hyperion Profitability and Cost Management database
name	nvarchar (80)	varchar2(80 CHAR)	Name of the selected Source stage
description	nvarchar (255)	varchar2(255 CHAR)	Description of the purpose for the selected stage
stage_order	integer	number (38,0)	The sequence position in which the selected stage is to be used within the model while modeling (assignments and rules creation), in calc script generation and calculation.
stage_prefix	nvarchar (80)	varchar2(80 CHAR)	Differentiating prefix for the stage name.
driver_dim_name	nvarchar (80)	varchar2(80 CHAR)	Name of the selected driver dimension
Is_intra	nvarchar (1)	varchar2(1 CHAR)	Flag determining whether intrastage assignment are allowed: • "Y" (Yes) specifies that intrastage assignments within the model stage are allowed. • "N" (No) specifies that intrastage 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.

Table 14-1 (Cont.) HPM_EXPD_STAGE

Field	SQL Server Data Type	Oracle Data Type	Description
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
dim21_name	nvarchar (80)	varchar2(80 CHAR)	Name of a dimension in the Destination stage
dim22_name	nvarchar (80)	varchar2(80 CHAR)	Name of a dimension in the Destination stage
dim23_name	nvarchar (80)	vvarchar2(80 CHAR)	Name of a dimension in the Destination stage
dim24_name	nvarchar (80)	varchar2(80 CHAR)	Name of a dimension in the Destination stage
dim25_name	nvarchar (80)	varchar2(80 CHAR)	Name of a dimension in the Destination stage

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 14-2 HPM_EXPD_POV

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

HPM_EXPD_DRIVER

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

Table 14-3 HPM_EXPD_DRIVER

Field	SQL Server Data Type	Oracle Data Type	Description
application_name	nvarchar (80)	varchar2(80 CHAR)	Name of the selected application
id	identity	integer(38,0)	Unique record ID
name	nvarchar (80)	varchar2(80 CHAR)	Name of the selected driver
description	nvarchar (255)	varchar2 (255 CHAR)	Description of the purpose for the selected driver
display_order	integer(38,0)	integer(38,0)	Display position of the driver within the list of all drivers in the model
formula	nvarchar (4000)	varchar2 (4000 CHAR)	Formula created for the driver by a user. This formula must be created using SQL syntax, and mathematically correct operations.

Table 14-3 (Cont.) HPM_EXPD_DRIVER

Field	SQL Server Data Type	Oracle Data Type	Description
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.
priority	integer	number (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 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.
volume_formula	nvarchar (4000)	varchar2(4000 CHAR)	For Rate-Based drivers only, the formula which has both the Rate and Volume formulas defined.
dst_measure_member_name	nvarchar (2000)	varchar2(2000 CHAR)	Name of the measure on the Destination table
tdv_measure_member_name	nvarchar (80)	varchar2(2000 CHAR)	Measure that is used as the denominator in the allocation formula: `Driver Value/TotalDriverValue` (`DV/TDV`)
operation_type	nvarchar (80)	varchar2(80 CHAR)	The operation type selected for the driver: • Rate-based Allocation • Ratio-based Allocation • Calculated Measures

HPM_EXPD_DRIVER_SEL

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

Table 14-4 HPM_EXPD_DRIVER_SEL

Field	SQL Server Data Type	Oracle Data Type	Description
application_name	nvarchar (80)	varchar2(80 CHAR)	Name of the selected application

Table 14-4 (Cont.) HPM_EXPD_DRIVER_SEL

Field	SQL Server Data Type	Oracle Data Type	Description
id	identity	integer(38,0)	Unique record ID
pov_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	Name of the first dimension member in the selected POV
pov_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)	Name of the second dimension member in the selected POV, if applicable
pov_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)	Name of the third dimension member in the selected POV, if applicable
pov_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)	Name of the fourth dimension member in the selected POV, if applicable
driver_dim_member_name	nvarchar (80)	varchar2(80 CHAR)	Name of the selected driver dimension
driver_name	nvarchar (80)	varchar2(80 CHAR)	Name of the selected driver

HPM_EXPD_DRIVER_EXCEP

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

Table 14-5 HPM_EXPD_DRIVER_EXCEP

Field	SQL Server Data Type	Oracle Data Type	Description
application_name	nvarchar (80)	varchar2(80 CHAR)	Name of the selected application
id	identity	integer(38,0)	Unique record ID
pov_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	Name of the first dimension member in the selected POV
pov_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)	Name of the second dimension member in the selected POV, if applicable
pov_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)	Name of the third dimension member in the selected POV, if applicable
pov_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)	Name of the fourth dimension member in the selected POV, if applicable
src_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	Name of the first dimension member in the source stage

Table 14-5 (Cont.) HPM_EXPD_DRIVER_EXCEP

Field	SQL Server Data Type	Oracle Data Type	Description
src_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)	Name of the second dimension member in the source stage, if applicable
src_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)	Name of the third dimension member in the source stage, if applicable
src_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)	Name of the fourth dimension member in the source stage, if applicable
src_dim5_member_name	nvarchar (80)	varchar2(80 CHAR)	Name of the fifth dimension member in the source stage, if applicable
driver_name	nvarchar (80)	varchar2(80 CHAR)	Name of the selected driver

HPM_EXPD_ASGN_RUL_SEL

The HPM_EXPD_ASGN_RULE_SEL view retrieves details about the assignment rule selections.

Table 14-6 HPM_EXPD_ASGN_RULE_SEL

Field	SQL Server Data Type	Oracle Data Type	Description
application_name	nvarchar (80)	varchar2(80 CHAR)	Name of the selected application
id	identity	integer(38,0)	Unique record ID
pov_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	Name of the first dimension member in the selected POV
pov_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)	Name of the second dimension member in the selected POV
pov_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)	Name of the third dimension member in the selected POV
pov_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)	Name of the fourth dimension member in the selected POV
src_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	Name of the first dimension member in the source stage
src_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)	Name of the second dimension member in the source stage
src_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)	Name of the third dimension member in the source stage
src_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)	Name of the fourth dimension member in the source stage

Table 14-6 (Cont.) HPM_EXPD_ASGN_RULE_SEL

Field	SQL Server Data Type	Oracle Data Type	Description
src_dim5_member_name	nvarchar (80)	varchar2(80 CHAR)	Name of the fifth dimension member in the source stage
rule_name	nvarchar (80)	varchar2(80 CHAR)	Name of the rule controlling the selected assignment. The rule must be present in the target database.

HPM_EXPD_CALCRULE_SNGLSRC

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

Table 14-7 HPM_EXPD_CALCRULE_SNGLSRC

Field	SQL Server Data Type	Oracle Data Type	Description
application_name		VARCHAR2(80 CHAR)	Name of the selected application
id	identity	NUMBER(38,0)	Unique identifier for each row. This can come from any sequence generator.
name	nvarchar (80)	VARCHAR2(80 CHAR)	Name of the calculation rule.
description	nvarchar (255)	varchar2(255 CHAR)	Long description of the calculation rule.
pov_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	Name of the first dimension member in the selected POV
pov_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)	Name of the second dimension member in the selected POV
pov_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)	Name of the third dimension member in the selected POV
pov_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)	Name of the fourth dimension member in the selected POV
calculation_sequence	integer	NUMBER(4,0)	Numeric value designating the relative order in which the calculation rule will be processed during calculation. This must be > 0 or you will get a check constraint error during insert.
tag	nvarchar (255)	VARCHAR2(255)	Specify search tags for the calculation rule in this column.
enabled_flag	nvarchar (1)	VARCHAR2(1 CHAR)	
first_sequence_priority	integer	NUMBER(38,0)	Designates the first driver sequence priority in the range of drivers to be executed for this rule. This must be > 0 or you will get a check constraint error during insert.

Table 14-7 (Cont.) HPM_EXPD_CALCRULE_SNGLSRC

Field	SQL Server Data Type	Oracle Data Type	Description
last_sequence_priority	integer	NUMBER(38,0)	Designates the last driver sequence priority in the range of drivers to be executed for this rule. This must be equal to or greater than *first_sequence_priority* or you will get a check constraint error during insert.
all_drivers_flag	nvarchar (1)	VARCHAR2(1 CHAR)	When `Y`, ignore values in the first_sequence_priority and last_sequence_priority fields during calculation.

HPM_EXPD_CALCRULE_CALCMSRS

The HPM_EXPD_CALCRULE_CALCMSRS view retrieves details about the calculated measures rule.

Table 14-8 HPM_EXPD_CALCRULE_CALCMSRS

Field	SQL Server Data Type	Oracle Data Type	Description
application_name	nvarchar (80)	VARCHAR2(80 CHAR)	Name of the selected application
id	identity	NUMBER(38,0)	Unique identifier for each row. This can come from any sequence generator.
name	nvarchar (80)	VARCHAR2(80 CHAR)	Name of the calculation rule.
description	nvarchar (255)	varchar2(255 CHAR)	Long description of the calculation rule.
pov_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	Name of the first dimension member in the selected POV
pov_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)	Name of the second dimension member in the selected POV
pov_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)	Name of the third dimension member in the selected POV
pov_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)	Name of the fourth dimension member in the selected POV
calculation_sequence	integer	NUMBER(4,0)	Numeric value designating the relative order in which the calculation rule will be processed during calculation. This must be > 0 or you will get a check constraint error during insert.
tag	nvarchar (255)	VARCHAR2(255)	Specify search tags for the calculation rule in this column.
enabled_flag	nvarchar (1)	VARCHAR2(1 CHAR)	

Table 14-8 (Cont.) HPM_EXPD_CALCRULE_CALCMSRS

Field	SQL Server Data Type	Oracle Data Type	Description
driver_name1	nvarchar (1)	VARCHAR2(1 CHAR)	Driver name to be associated with this calculation rule.
driver_name2	nvarchar (80)	VARCHAR2(80 CHAR)	Driver name to be associated with this calculation rule.
driver_name3	nvarchar (80)	VARCHAR2(80 CHAR)	Driver name to be associated with this calculation rule.
driver_name4	nvarchar (80)	VARCHAR2(80 CHAR)	Driver name to be associated with this calculation rule.
driver_name5	nvarchar (80)	VARCHAR2(80 CHAR)	Driver name to be associated with this calculation rule.
dest_assign_rule_name1	nvarchar (80)	VARCHAR2(80 CHAR)	Destination Assignment Rule to be associated with this calculation rule.
dest_assign_rule_name2	nvarchar (80)	VARCHAR2(80 CHAR)	Destination Assignment Rule to be associated with this calculation rule.
dest_assign_rule_name3	nvarchar (80)	VARCHAR2(80 CHAR)	Destination Assignment Rule to be associated with this calculation rule.
dest_assign_rule_name4	nvarchar (80)	VARCHAR2(80 CHAR)	Destination Assignment Rule to be associated with this calculation rule.
dest_assign_rule_name5	nvarchar (80)	VARCHAR2(80 CHAR)	Destination Assignment Rule to be associated with this calculation rule.

HPM_EXPD_CALCRULE_MULTISRC

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

Table 14-9 HPM_EXPD_CALCRULE_MULTISRC

Field	SQL Server Data Type	Oracle Data Type	Description
application_name	nvarchar(80)	VARCHAR2(80 CHAR)	Name of the selected application
id	identity	NUMBER(38,0)	Unique identifier for each row. This can come from any sequence generator.
name	nvarchar (80)	VARCHAR2(80 CHAR)	Name of the calculation rule.
description	nvarchar (255)	varchar2(255 CHAR)	Long description of the calculation rule.
pov_dim1_member_name	nvarchar (80)	varchar2(80 CHAR)	Name of the first dimension member in the selected POV
pov_dim2_member_name	nvarchar (80)	varchar2(80 CHAR)	Name of the second dimension member in the selected POV
pov_dim3_member_name	nvarchar (80)	varchar2(80 CHAR)	Name of the third dimension member in the selected POV

Table 14-9 (Cont.) HPM_EXPD_CALCRULE_MULTISRC

Field	SQL Server Data Type	Oracle Data Type	Description
pov_dim4_member_name	nvarchar (80)	varchar2(80 CHAR)	Name of the fourth dimension member in the selected POV
calculation_sequence	integer	NUMBER(4,0)	Numeric value designating the relative order in which the calculation rule will be processed during calculation. This must be > 0 or you will get a check constraint error during insert.
tag	nvarchar (255)	VARCHAR2(255)	Specify search tags for the calculation rule in this column.
enabled_flag	nvarchar (1)	VARCHAR2(1 CHAR)	
driver_name	nvarchar (80)	VARCHAR2(80 CHAR)	Driver name to be associated with this calculation rule.
src_assign_rule_name1	nvarchar (80)	VARCHAR2(80 CHAR)	Source assignment rule to be associated with this calculation rule.
src_assign_rule_name2	nvarchar (80)	VARCHAR2(80 CHAR)	Source assignment rule to be associated with this calculation rule.
src_assign_rule_name3	nvarchar (80)	VARCHAR2(80 CHAR)	Source assignment rule to be associated with this calculation rule.
src_assign_rule_name4	nvarchar (80)	VARCHAR2(80 CHAR)	Source assignment rule to be associated with this calculation rule.
src_assign_rule_name5	nvarchar (80)	VARCHAR2(80 CHAR)	Source assignment rule to be associated with this calculation rule.
dest_assign_rule_name1	nvarchar (80)	VARCHAR2(80 CHAR)	Destination assignment rule to be associated with this calculation rule.
dest_assign_rule_name2	nvarchar (80)	VARCHAR2(80 CHAR)	Destination Assignment Rule to be associated with this calculation rule.
dest_assign_rule_name3	nvarchar (80)	VARCHAR2(80 CHAR)	Destination assignment rule to be associated with this calculation rule.
dest_assign_rule_name4	nvarchar (80)	VARCHAR2(80 CHAR)	Destination assignment rule to be associated with this calculation rule.
dest_assign_rule_name5	nvarchar (80)	VARCHAR2(80 CHAR)	Destination assignment rule to be associated with this calculation rule.

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 Oracle Hyperion Profitability and Cost Management 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\O
racle
```

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.
2. Select the connection for the Oracle Fusion Performance Management Product Schema.
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. Standard

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.

A

Application Design Best Practices

Managing Application Scalability

Allocation processes are capable of expanding the size of the initial data many times over.

Allocation rules take the original data values at the leaf level and split them across all destination cells. It is not unusual to find that, after executing allocation rules, your data has increased to hundreds of times its original size. A waterfall of allocation rules can split these smaller amounts into many smaller values.

Thinking about the data scale early in the design process is required to avoid dealing with millions of microscopic data points at the end of the model calculations. Expanded data can severely complicate integrating results from Oracle Hyperion Profitability and Cost Management with downstream reporting systems or generating data extracts for sending back to the GL.

Dimension Count

The best means of keeping data growth under control is to consider factors early in the design phase that will impact data scale, especially in a post allocated state. The number of dimensions used in the application is the first scalability consideration.

Be cautious about adding more dimensions to the data without a solid reason. Data growth in Oracle Hyperion Profitability and Cost Management is mostly impacted by the number of splits of data into smaller and smaller values. Before adding new dimensions, verify that a new physical dimension is required. First time Profitability and Cost Management designers commonly adopt all of the dimensions present in the source data simply because they are there and might, someday, be needed.

Consider if the dimension is required for either *reporting final results* or *differentiating data in order to support an allocation process* . If neither of these are true, you should strongly consider eliminating the dimension.

If the additional dimension is really an alternate expression of an existing dimension, consider using an alternate hierarchy or attribute dimension instead. This will provide the means for reporting on the desired categories without increasing data size.

While limiting dimensions is strongly advised, adding a dimension for future growth is a good idea. As long as the dimension is only using a single "nomember" selection in all model artifacts, the "spare" dimension will have little impact on performance.

Dimension Member Count

Data scale in Oracle Hyperion Profitability and Cost Management is a function of the number of dimensions and the size (member count) of those dimensions. The range sizes for rules are impacted by both *dimension count* and *member count*.

When designing the application, it is wise to be judicious in the use of very large dimensions. Profitability and Cost Management applications work best when the dimensions are fewer and

smaller. This is not to say that Profitability and Cost Management cannot handle large and sparse dimensions, but that using many large dimensions can cause performance issues in calculation once you start design rules to work across the vast and sparse ranges that are caused by using many large dimensions.

Use the same rationale for member detail within a dimension as you do for choosing dimensions in the first place:

- Will the detail be needed for reporting?
- Is it needed to differentiate data to support allocation logic?

If the answer to both questions is no, you should consider using a more aggregated level of members in the dimension. For example, use Cost Pools instead of the lowest level account or Product Family instead of SKU.

Attributes and UDAs

Attributes are powerful tools for both reporting and allocation logic; however, the management of the associations between base members and Attribute members can sometimes become cumbersome. Using UDAs instead of Attributes is advisable when you are working with very large base dimensions (10,000+) or a large number of unique attribute values.

Data Scale Management

Being aware of the possible negative impacts of uncontrolled data growth early in the design cycle is important. It is important to resist the urge to add every last bit of detail to the application in the quest to support any reporting need that might ever occur. Know the reporting requirements of the application before you start designing it. This is the only way the designer will be able to answer the question "do we really need that level of detail….?"

With this knowledge in hand, there are two key design methods for limiting growth:

- Summarize Starting Data
- Summarize Allocation Midpoints

Summarize Starting Data

Not all data is truly needed. As discussed earlier, the use of summarization points in data *before* it is loaded is one of the best tools. Account detail is a common example. Rather than loading expenses at the lowest level of detail, use aggregate cost pools instead. Use this strategy for every dimension where possible in your data. Refer to the earlier questions regarding detail needed for reporting or allocation process.

Ask these questions to determine whether details are needed for the reporting or allocation process:

- Is the detail needed for reporting?
- Is the detail needed to differentiate data to support allocation logic?

This step alone can shrink starting data size by one or more orders of scale.

Summarize Allocation Midpoints

Similar to starting data points, there are natural funnel points in an application where detail levels used in previous allocations step are no longer required. In these cases, using a pooling

strategy in the middle stages of an application can help reshrink data down to a more manageable size.

This is especially useful in a stage immediately prior to allocating financial data out to market facing dimensions like Customer, Region/Territory, Product/SKU. Market facing dimensions are usually the largest dimensions in an application, and are the stage where data explosion commonly happens.

If you can slim down the data scale after operational center allocations, you will have fewer data points to allocate across market intersections, and the end result will be a smaller data set.

Managing Dimension Design

Dimension design can have a large impact on reporting and calculation performance.

Oracle Hyperion Profitability and Cost Management uses an Oracle Essbase Aggregate Storage (ASO) Database. ASO databases have different performance characteristics and optimization methods than the block storage (BSO) databases used by other Oracle EPM business processes.

Dimension Types

Oracle Essbase ASO offers three dimension types:

- Stored
- Dynamic
- Hierarchies Enabled

Stored

In general, stored dimensions offer the best performance for both calculation and reporting.

Stored Dimensions Support

- Fastest aggregation
- Shared members
- Attributes and UDAs

Limitations

- Members may only use the Addition (+) consolidation operator
- No member formulas
- Shared members must appear after the base member of the share

Dynamic

Dynamic dimensions offer the most design flexibility, but at the cost of limiting calculation and reporting performance.

Dynamic Dimensions Support

- Any consolidation operator (+, -, *, /, ~)

- Member formula
- Shared members
- Attributes and UDAs

Dynamic Dimensions Limitations

- Slowest aggregation

  This has an impact on both calculation and reporting performance. Using multiple dynamic dimensions in an application compounds this problem.
- Shared members must appear after the base member of the share

Hierarchies Enabled

Hierarchies Enabled dimensions offer a hybrid of Stored and Dynamic hierarchies that allow parts of the dimension to act and perform similar to Stored dimensions and other hierarchies to act and perform similar to Dynamic dimensions

Hierarchies Enabled Dimensions Support

- A combination of multiple Stored and Dynamic Hierarchies
- Stored hierarchies have the same feature support and limitations as the Stored Dimension type.
- Dynamic hierarchies have the same feature support and limitations as the Dynamic Dimension type.

Hierarchies' Enabled Dimension Limitations

- The first Generation 2 member hierarchy must be a Stored hierarchy type.
- The root member must have data storage of *Label Only*

Hierarchy Top Levels

The top levels of dimensions can have an impact on usability in reporting tools and in Oracle Hyperion Profitability and Cost Management screens like Rule Balancing that use a Model View. In general, setting up the dimension so that data rolls up to the root member naturally will make the dimension easier to use in reports.

Rollup and Reporting

Whenever possible, ensure that all data in the main hierarchy rolls up to the top member in the dimension. This enables users to skip making an explicit member selection in reports if they want to see *all* data for that dimension. This also simplifies the use of Rule Balancing and other reporting tools used in an ad hoc fashion.

No Member

A common approach to mixing data with irregular dimensionality is to use members that designate the data point to not have a reference in the dimension. Using a "No" member such as "NoProduct" supports storing data that has not yet been designated to a specific product.

When using a "No" member in a Hierarchies Enabled dimension, it is advisable to place the "No" member in the 1st generation 2 hierarchy. This helps simplify reporting by ensuring that all

data points for the dimension, including data not yet attributed in that dimension, aggregate to the top of the dimension.

Best Practice Example

Using a Product hierarchy as an example:

Single Hierarchy Dimension

Dimension Root Member (Product) Data Storage: Store Data

1st Generation 2 Hierarchy (Total Product) Aggregation Operator: Addition

- 1st Child (All Products) Aggregation Operator: Addition. The rest of the Product hierarchy is placed under this member.
- 2nd Child (NoProduct) Aggregation Operator: Addition. This is the member used if data points don't have an associated Product member.

Multiple Hierarchy Dimension

Dimension Root Member (Product) Data Storage: Label Only

- 1st Generation 2 Hierarchy (Total Product) Aggregation Operator: Addition.
  - 1st Child (All Products) Aggregation Operator: Addition. The rest of the Product hierarchy is placed under this member.
  - 2nd Child (NoProduct) Aggregation Operator: Addition. This is the member used if data points don't have an associated Product member.
- 2nd Generation 2 Hierarchy (Alternate Hierarchy) Aggregation Operator: Ignore.
- 3rd Generation 2 Hierarchy (Alternate Hierarchy) Aggregation Operator: Ignore.
- …Nth Generation 2 Hierarchy (Alternate Hierarchy) Aggregation Operator: Ignore.

Duplicate Members

Duplicate members are base members (not shared members) that store data and have identical names. Oracle Hyperion Profitability and Cost Management supports duplicate members across different dimensions but not duplicate members within the same dimension.

Duplicate dimension members are useful in cases where costs are tracked as they move from a source organization unit to a destination organizational unit, and it is necessary to report on the costs moved between the sources and targets.

While Profitability and Cost Management supports duplicate members, there are challenges created by using them. In Oracle Essbase, the data cells containing data must be uniquely identified. When no duplicate members are used, the system uses the member name combination to identify a cell.

With duplicate members, a further qualification is required. For example, instead of referring to a data cell organization member (for example, "Marketing") the reference must also include the dimension (for example, "Entity.TotalEntitiy.Administration.Marketing"). The finer qualification required for duplicate members can cause issues when creating reports and modifying dimensions, particularly in moving members. When a duplicate member is moved, its fully qualified name has changed. This causes the Database Redeploy process to be unable to find a proper cell for data after modifying the dimension, and will result in dropped data values.

Rather than creating duplicate dimensions, you should prefix the members of one or both dimensions to avoid creating duplicates.

Managing Application Logic Design

Create a map of your application design before creating rules.

Related Topics

- Simple Strategies for Managing Rule Count
- Allocation Rule Types and Options
- Rule Ranges
- Using Parallel Execution in Rulesets
- Custom Calculation Rules

Simple Strategies for Managing Rule Count

It is best to create a map of your application design before beginning to create rules. Many users dive into replicating allocations from a legacy system before looking at the total scope of allocation rules and trying to rationalize the entire process. In many cases, if you can take a fresh look at the process and consider many of the tools that Oracle Hyperion Profitability and Cost Management provides, you can find ways to reduce steps in the process and make the process easier to understand – and easier to explain to others.

- Identify Where Common Logic Can be Applied Using a Single Rule
- Use the Absence of Driver Data as an Allocation Filter

Identify Where Common Logic Can be Applied Using a Single Rule

This is usually the place where migrations of legacy allocation rules to Oracle Hyperion Profitability and Cost Management first encounter challenges and where there are easy solutions. If you identify allocation patterns that are common among many departments or entities, it is likely that you can combine many steps from a legacy process into a single rule in Profitability and Cost Management.

Check for cases where many areas of your logic require allocation to the same set of destinations using the same driver value. In these cases, the Same as Source option on the Destination tab will allow you to run many parallel allocation steps in a single rule.

Use the Absence of Driver Data as an Allocation Filter

In cases where members in your destination range that should receive an allocation are scattered among members that should not receive an allocation, keep in mind that just because a member is part of the destination it does not automatically receive an allocated amount. The allocations are only sent to destinations that have a driver value.

Using this behavior of the allocation engine to your advantage, you can create a single rule that crosses a wider range of sources and destinations but will only allocate to the intersections where it finds driver data. This allows you to create a single rule, where in a legacy system, you may have had to create many rules.

Allocation Rule Types and Options

The allocation relationships in each rule impact the methods that Oracle Hyperion Profitability and Cost Management uses to construct calculation logic and how the Oracle Essbase engine executes it. Not all rule types are equal in terms of performance.

The major rule types in order of performance are:

- Simple Dimension Extension
- Complex Dimension Extension
- Simple Reclassification
- Complex Reclassification
- Same as Source, Different Dimension

Simple Dimension Extension

- Every dimension's source and destination relationship is either set to same as source, or is a one source (a single level 0 source) to many destinations relationship.
- Every driver member selection is level 0.
- Every destination dimension is a single member selection, either level 0 or a parent.
- The Simple Dimension Extension rule type is the best performing type.

Complex Dimension Extension

- Every dimension's source and destination relationship is either set to same as source, or is a one source (a single level 0 source) to many destinations relationship.
- Any driver member selection is a parent member.
- Any destination dimension has multiple distinct member selections.

Simple Reclassification

One dimension's source and destination relationship is a many-to-many relationship, whether or not the *many* members is defined by multiple distinct member selections or by the selection of a parent member. The rest of the dimension source and destination relationships are same-as- source.

Complex Reclassification

Multiple dimension's source and destination relationships are a many-to-many relationship, whether or not the *many* members is defined by multiple distinct member selections or the selection of a parent member. The rest of the dimension source and destination relationships are same-as-source.

Complex Reclassification rules are usually the slowest performing rule type.

Same as Source, Different Dimension

One or more of the dimension's source and destination relationships use the same as source option where the source and destination dimensions are different.

This option can be used with any of the allocation rule types, and will add complexity to the execution and reduce performance.

Use this option carefully and do frequent performance benchmarks during application build to determine if the impacts are acceptable.

Allocation Rule and Member Selection Options Allowed for Each Rule Type

Table A-1 Allocation Rule and Member Selection Options Allowed for Each Rule Type

	Source Dimensions	Destination Dimensions	Driver Dimensions
Simple Dimension Extension (SDE)	• A single level 0 member • Any other member selection if Destination is Same as Source for the same dimension	• At least one Same as Source • A single level 0 member • Any other member selection if no Driver selection made for same dimension	• Allocate Evenly • Specified Driver Location is a Level 0 Tuple
Complex Dimension Extension (CDE)	• A single level 0 member • Any other member selection if Destination is Same as Source for the same dimension	• At least one Same as Source • A single level 0 member • Any other member selection	• Allocate Evenly • Specified Driver Location is a Level 0 Tuple • Specified Driver Location is not a Level 0 Tuple
Simple Reclassification (SR)	• A single level 0 member • Any other member selection	• Same as Source • A single level 0 member • Any other member selection if no Driver selection made for same dimension	• Allocate Evenly • Specified Driver Location is a Level 0 Tuple
Complex Reclassification (CR)	• A single level 0 member • Any other member selection	• Same as Source A single level 0 member • Any other member selection	• Allocate Evenly • Specified Driver Location is a Level 0 Tuple • Specified Driver Location is not a Level 0 Tuple
Same as Source, Different Dimension (SAD)	• A single level 0 member • Any other member selection	• Same as Source • Same as Source, Different Dimension • Any other member selection	• Allocate Evenly • Specified Driver Location is a Level 0 Tuple • Specified Driver Location is not a Level 0 Tuple

Rule Ranges

Another aspect of rule design that impacts performance are source and destination or target ranges. Careful management of the rule ranges can boost performance.

Range Size and Sparsity

In general, the larger the source, destination, or target range is, the longer the rule takes to execute. Some rules are able to execute across extremely large ranges, but some are not. The rule type has a larger impact on calculation time; however, for a given rule/rule type, a smaller source and destination range executes faster. Be wary of creating rules with extremely large ranges. You can review ranges on the Rule Definition screen or in the Execution statistics report.

Using Source Segmentation

One option for reducing source range is to split a rule into several smaller rules by using the same destination and driver basis definitions, and splitting the source range into smaller parts. You may do this manually using several rules, or you can use the Source Segmentation option on the rule's source tab to have the system split execution into smaller segments automatically.

Using Filters or Alternate Hierarchies

In some cases, the members that need to be included in either the source or the destination are not naturally housed together in the native hierarchy. Using alternate hierarchies to group members in a way that makes them more "rule friendly" is one way to limit rule ranges and make rule logic easier to understand. Another option is using attribute filters.

Avoiding Using Negative Driver Data Values

Driver data with negative values can create challenges for the allocation engine that can result in unexpected and unusable results. The allocation engine uses the driver values found at each destination location as a numerator in the ratio Driver Value / Sum of All Driver Values. This allocation ratio is multiplied by the source value to determine the *AllocationIn* amount to apply to the destination. Using a mix of positive and negative driver values creates cases where the denominator is no longer accurate and can cause inflation and deflation of *AllocationIn* amounts in the rule results. These results are very difficult to explain and justify, so it is best practice to avoid this situation entirely.

Avoiding Using Dynamic Members as Drivers

If a driver value requires a formula, it is best to calculate the value before allocation rules are executed. During rule execution, every possible destination location is reviewed by the rule to determine what the driver value is at that location. If a calculated member is referenced as the driver, then the system must execute the member formula for every possible location. This can result in potentially billions of unnecessary computations which severely impact rule performance.

A better solution is to prepare driver data for the rule by preceding it with a custom calculation rule to execute the formula needed and *store* the driver value in a stored member. This allows the allocation rule to execute normally and quickly skip locations without any driver data.

Using Parallel Execution in Rulesets

In some cases, executing rules in a parallel fashion can help overall performance. Calculation options are set at the Ruleset level and affect all rules in the Ruleset. The default option for all Rulesets is s*equential processin*g. This means that every rule is executed individually, and all rules in the Ruleset are executed one by one.

Ensure that Your Rules are Parallel Safe

Rules often have an impact on the execution of other rules. If one rule allocates into the source range of another rule, the sequence of running the rules can have a dramatic impact on the final result. This is why the default calculation option is sequential.

If there are rules that can safely be executed in parallel, then using the parallel calculation option can help improve performance. Set the parallel processing option on the Ruleset or Rulesets you want to enable. You also need to set the Maximum Concurrent Threads option on the Application Properties screen.

Experiment with the Number of Threads

Finding the right level of concurrent threads is a matter of experimentation. There is a point at which performance improvements cease, and the use of additional threads to launch additional rules is counterproductive. The best practice for parallel calculation is to test Rulesets starting with four concurrent threads and gradually increase the number of threads until performance improvements stop. While rules that do not mix source ranges are good candidates for parallel processing, their destination ranges may still share common locations. In these cases, Oracle Essbase will still have to pause one rule until the other one has finished writing destination data. For this reason, you will see a point at which increasing the thread count no longer helps with performance.

Custom Calculation Rules

Custom Calculation Rules are impacted by large target ranges in the same way as Allocation rules. In general, smaller ranges execute faster. In almost all cases, using the `NONENPTYTUPLE` statement in your rule formula helps performance, especially if the target range for the rule is very large.

B

Importing Data into Profitability and Cost Management

About Importing Data

You can enter data directly into Oracle Hyperion 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 Detailed Profitability applications using a set of import staging tables and import configurations.

Model data is imported from several sources:

- 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 data and applications can be imported using Oracle Hyperion Enterprise Performance Management System 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 Oracle Hyperion Enterprise Performance Management Workspace and Oracle 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 Oracle Hyperion 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 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:

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

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 several 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.

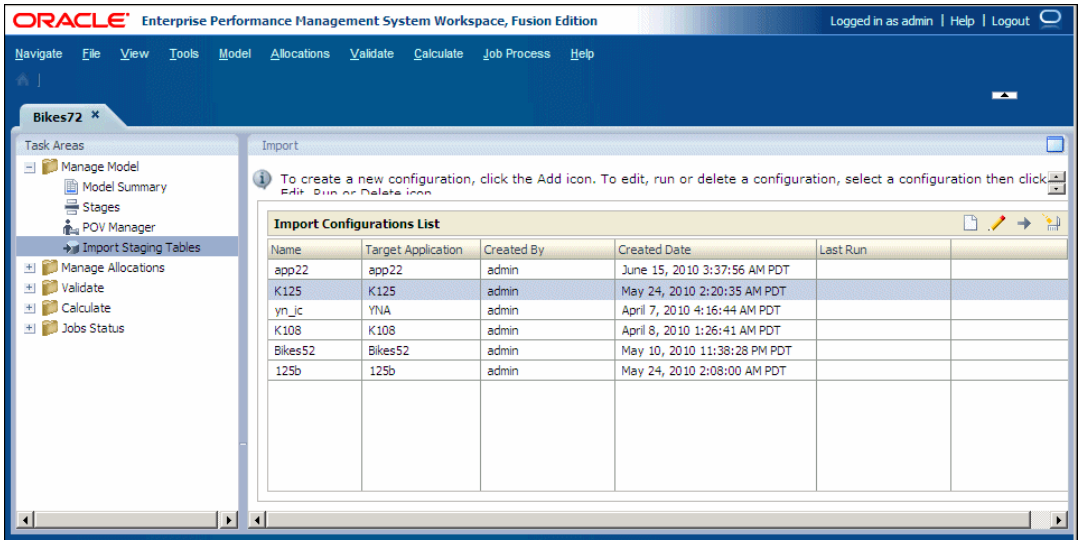
   There are no naming restrictions for the database.

> **Caution:**
>
> Oracle recommends that, before importing data or artifacts, you create a backup directory of your databases in Oracle Hyperion Enterprise Performance Management Workspace and Oracle Essbase. Contact your administrator for assistance.

2. Run the appropriate script against the source database to automatically create the staging tables:

   For Detailed Profitability applications, run the `create_dp_staging.sql` script. The tables are generated using the schema provided in 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 Oracle Hyperion Profitability and Cost Management model.
7. From `Task Areas`, select **Manage Model**, then **Import Staging Tables**.



8. 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.

9. 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.

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

    Step 2 of the Import Data dialog box is displayed.
12. Under **Configuration**, select the **Target Application**.

    The target application is the Profitability and Cost Management application into which the data is being imported.
13. 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.
14. Click **Next**.

    Step 3 of the Import Data dialog box is displayed.
15. 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.
16. 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.
17. **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.

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.

Deleting Import Configurations

You can delete import configurations.

To delete import configurations:

1. From **Task Areas**, select **Manage Model**, and 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.

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 Oracle Hyperion 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.

C

Migrating Data Using EPM System Lifecycle Management

About Lifecycle Management

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

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

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

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

For detailed information about Lifecycle Management and Oracle Hyperion Profitability and Cost Management artifacts for Detailed Profitability, see the *Oracle Enterprise Performance Management System Lifecycle Management Guide*.

ApplicationData and Dimensions Artifacts for Management Ledger

For Management Ledger Applications, LCM Export includes a Metadata folder that contains these artifacts:

- ApplicationData
- Dimensions

These artifacts reside in the following folder structure, respectively:

- resource\Metadata\ApplicationData\ — only one file here
- resource\Metadata\Dimensions\ — one file for each dimension

For example:

```
resource\Metadata\ApplicationData\application.xml
resource\Metadata\Dimensions\Account.txt
resource\Metadata\Dimensions\Product.txt
resource\Metadata\Dimensions\Year.txt
```

The ApplicationData artifact type is for future use and is not currently available for export or reimport. For correct operation of LCM Export, you must deselect ApplicationData when exporting or importing any dimensions.

To import Dimensions, you must first create a new empty application in the Profitability Applications Console. Dimensions can only be imported into an empty application that has not been enabled.

> **Note:**
>
> For these artifacts, the artifact type is the same as the artifact name.

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>
```

```
SetHandler weblogic-handler

PathTrim

KeepAliveEnabled ON

KeepAliveSecs 20

WLIOTimeoutSecs 3000

WeblogicCluster servername:6756
```

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\htppd.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.

D

Backing Up Profitability and Cost Management Components

There are several Oracle Hyperion 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

> **Note:**
>
> Depending on application type, you may not use some of these databases.

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

Table 1, 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 D-1 Profitability and Cost Management Database Backup Requirements

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

Table D-1 (Cont.) Profitability and Cost Management Database Backup Requirements

Database	Database Type	Backup Options
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*.

E

Essbase Naming Conventions

Generated Calculation Script Naming Conventions

When Oracle Hyperion Profitability and Cost Management generates Oracle 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 E-1 Calculation Script Suffixes

Script Type	Cost Layer	Revenue Layer
Interstage Allocation	"a"	"r"
Intrastage Allocation	"i"	"t"

- 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), Oracle Essbase does not automatically convert it to upper- and lowercase (for instance, Newapp).

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

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

- Use no more than 8 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:

- 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 Oracle Essbase.
- Do not use forward slashes in member names.
- For time periods in custom calendars, do not use spaces in prefixes.
- Do not use 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
  - EMPTYPARM
  - EQ
  - EQOP
  - EXCEPT
  - EXP
  - EXPERROR

- FLOAT
- FUNCTION
- GE
- GEN
- GENRANGE
- GROUP
- GT
- ID
- IDERROR
- INTEGER
- LE
- LEVELRANGE
- LOOPBLOCK
- LOOPPARMS
- LT
- MBR
- MBRNAME
- MBRONLY
- MINUS
- MISSING
- MUL
- MULOP
- NE
- NON
- NONINPUT
- NOT
- OR
- PAREN
- PARENPARM
- PERCENT
- PLUS
- RELOP
- SET
- SKIPBOTH
- SKIPMISSING
- SKIPNONE
- SKIPZERO
- TO

- TOLOCALRATE
- TRAILMISSING
- TRAILSUM
- UMINUS
- UPPER
- VARORXMBR
- XMBRONLY
- $$UNIVERSE$$
- #MISSING
- #MI

Essbase Naming Conventions for Attribute Calculations Dimension

The members that Oracle Essbase creates in the attribute calculations dimension (Sum, Count, Min, Max, and Avg) are not considered reserved words.

This is 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, 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.