

Oracle® Data Relationship Management Suite

Integrating Oracle Data Relationship
Management Suite with Enterprise
Performance Management



Release 11.2.0
F21887-02
December 2019

ORACLE®

Oracle Data Relationship Management Suite Integrating Oracle Data Relationship Management Suite with Enterprise Performance Management, Release 11.2.0

F21887-02

Copyright © 2015, 2020, Oracle and/or its affiliates.

Primary Author: EPM Information Development Team

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or "commercial computer software documentation" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Documentation Accessibility

Documentation Feedback

1 About Data Relationship Management Suite

2 Migrating Performance Management Architect Applications to Data Relationship Management

Preparing Performance Management Architect Application Metadata	2-1
Exporting Planning Metadata from Performance Management Architect	2-2
Exporting Financial Management Metadata from Performance Management Architect	2-2
Exporting Profitability and Cost Management Standard and Detailed Application Metadata from Performance Management Architect	2-3
Exporting Essbase Metadata from Performance Management Architect	2-4
Importing Performance Management Architect Application Metadata into Data Relationship Management	2-5
Exporting from Data Relationship Management and Importing to EPM Applications	2-6

3 Using Data Relationship Management with Oracle Hyperion Planning and EPM Cloud Planning

Configuring Planning Metadata in Data Relationship Management	3-2
Locating and Loading the Application Template	3-2
Defining Planning Applications	3-2
Defining and Configuring Planning Dimensions	3-3
Dimension to Hierarchy Organization	3-4
Configuring Planning Dimension Types	3-4
Enabling Node Types	3-5
Configuring Oracle Hyperion Planning Export Profiles	3-5
Optimal Configurations	3-6

Initializing Data Relationship Management with Planning Dimensions	3-11
Managing Planning Dimension Members in Data Relationship Management	3-13
Managing Planning Dimension Membership	3-14
Managing Planning Dimension Member Properties	3-15
Validating Planning Dimension Members and Properties	3-15
Exporting Dimension Members from Data Relationship Management	3-15
Importing Dimension Members into Oracle Hyperion Planning	3-16
Importing Dimension Members into EPM Cloud Planning	3-17

4 Using Data Relationship Management with FDMEE

Integration	4-1
Prerequisites	4-2
Additional Documentation	4-2
Configuring the FDMEE System Environment	4-2
Configuring FDMEE Metadata in Data Relationship Management	4-3
FDMEE Application Template	4-3
Planning Application Template Metadata	4-5
Locating and Loading the Application Template	4-5
Configuring the External Connections to the FDMEE Database	4-7
Reconnecting Imports and Exports to the FDMEE Database	4-8
Configuring Domains	4-9
Configuring Blend After Import	4-11
Extending Imports to Use Additional Custom Top Member and Aggregation	4-12
Adding Mapping Validations	4-12
Configuring the FDMEE Template for Custom Data Relationship Management Implementations	4-12
Imports	4-13
Exports	4-14
Loading GL Values and Hierarchies into Data Relationship Management	4-15
Mapping ERP to EPM Members in Data Relationship Management	4-16
Setting the FDMEE_MAP_VERSION Variable	4-16
Performance Management Architect and Planning Dimensions Used for Mappings	4-16
Mapping ERP to EPM Members in Data Relationship Management	4-18
Exporting Mappings to FDMEE	4-19

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, send email to epmdoc_ww@oracle.com, or, in an Oracle Help Center topic, click the Feedback button located beneath the Table of Contents (you may need to scroll down to see the button).

Follow EPM Information Development on these social media sites:

LinkedIn - http://www.linkedin.com/groups?gid=3127051&goback=.gmp_3127051

Twitter - <http://twitter.com/hyperionepminfo>

Facebook - <http://www.facebook.com/pages/Hyperion-EPM-Info/102682103112642>

YouTube - <https://www.youtube.com/oracleepminthecloud>

1

About Data Relationship Management Suite

Oracle Data Relationship Management Suite consists of:

- Oracle Data Relationship Management
- Oracle Data Relationship Management Read Only Access
- Oracle Data Relationship Steward
- Oracle Data Relationship Governance
- Oracle Data Relationship Management Analytics
- Oracle Data Relationship Management for Oracle Hyperion Enterprise Planning Suite
- Oracle Data Relationship Management for Oracle Hyperion Financial Close Suite

2

Migrating Performance Management Architect Applications to Data Relationship Management

In release 11.2.0, customers who previously used Oracle Hyperion EPM Architect to manage metadata such as hierarchy members, member relationship structures, and member property values can use Oracle Data Relationship Management instead.

To learn about Data Relationship Management, click [here](#).

In general, the steps to migrate metadata from an Performance Management Architect application to a Data Relationship Management application are:

1. Export your 11.1.2.4 application metadata from Performance Management Architect to a file using the EPMA File Generator Utility.
2. Update the file so that it's compatible with Data Relationship Management. This may be a file conversion utility or a manual procedure.
3. Create an application in Data Relationship Management using predefined templates and import your application metadata (the converted file from step 2) to it.
4. Manage your metadata in Data Relationship Management.
5. Export application metadata out of Data Relationship Management and import it into your Oracle Enterprise Performance Management System application.

Going forward, you'll maintain metadata in Data Relationship Management and export it to your application.

For detailed instructions on upgrading to Release 11.2.0, see [Upgrading EPM System](#).

Related Topics

- [Preparing Performance Management Architect Application Metadata](#)
- [Importing Performance Management Architect Application Metadata into Data Relationship Management](#)
- [Exporting from Data Relationship Management and Importing to EPM Applications](#)

Preparing Performance Management Architect Application Metadata

For all EPM applications, use the EPMA File Generator Utility (`EPMAFileGenerator.exe`) to create an export of your 11.1.2.4.x application metadata.

For more information on using the EPMA File Generator, see [From Performance Management Architect, Profitability and Cost Management, Essbase \(ASO\) and Essbase \(BSO\) Applications](#).

For steps specific to your EPM application, see:

- [Exporting Planning Metadata from Performance Management Architect](#)
- [Exporting Financial Management Metadata from Performance Management Architect](#)
- [Exporting Profitability and Cost Management Standard and Detailed Application Metadata from Performance Management Architect](#)
- [Exporting Essbase Metadata from Performance Management Architect](#)

Exporting Planning Metadata from Performance Management Architect

Export Metadata

To export 11.1.2.4 Oracle Hyperion Planning application metadata:

1. Launch the EPMA File Generator Utility by selecting **Start, Oracle EPM Systems, Foundation Services, Performance Management Architect**, and then **Start File Generator**.

The file generator is installed with Oracle Hyperion EPM Architect by default to:

```
C:\Oracle\Middleware\EPMSys11R1\products\Foundation\BPMA  
\EPMAFileGenerator\bin
```

2. Click **From Planning Application**.
3. Enter the following information:
 - User name and password for the EPMA administrator
 - URL for EPMA Web Server
 - Name of the EPMA application
 - EPMA import file
4. Click **Execute**.

Convert Export File

Convert the export file to an Oracle Data Relationship Management compatible import file. For information on converting the export file, go to [Oracle Support](#) and navigate to the white paper "How to Migrate Metadata from EPMA to DRM Application (Doc ID 2626317.1)". This white paper also contains a link to a conversion utility that you can use to convert your export file to a file that's compatible with Data Relationship Management.

For information on how to create imports for Data Relationship Management, see [Working with Imports](#).

Exporting Financial Management Metadata from Performance Management Architect

Export Metadata

To export 11.1.2.4 Oracle Hyperion Financial Management application metadata:

1. Launch the EPMA File Generator Utility by selecting **Start, Oracle EPM Systems, Foundation Services, Performance Management Architect**, and then **Start File Generator**.

The file generator is installed with Oracle Hyperion EPM Architect by default to:

```
C:\Oracle\Middleware\EPMSys11R1\products\Foundation\BPMA  
\EPMAFileGenerator\bin
```

2. Click **From HFM Application**.
3. Enter the following information:
 - User name and password for the EPMA administrator
 - URL for EPMA Web Server
 - Name of the EPMA application
 - EPMA import file
4. Click **Execute**.

Convert Export File

Convert the export file to an Oracle Data Relationship Management compatible import file. For information on converting the export file, go to [Oracle Support](#) and navigate to the white paper "How to Migrate Metadata from EPMA to DRM Application (Doc ID 2626317.1)". This white paper also contains a link to a conversion utility that you can use to convert your export file to a file that's compatible with Data Relationship Management.

For information on how to create imports for Data Relationship Management, see [Working with Imports](#).

Exporting Profitability and Cost Management Standard and Detailed Application Metadata from Performance Management Architect

Export Metadata

To export 11.1.2.4 Oracle Hyperion Profitability and Cost Management application metadata:

1. Launch the EPMA File Generator Utility by selecting **Start, Oracle EPM Systems, Foundation Services, Performance Management Architect**, and then **Start File Generator**.

The file generator is installed with Oracle Hyperion EPM Architect by default to:

```
C:\Oracle\Middleware\EPMSys11R1\products\Foundation\BPMA  
\EPMAFileGenerator\bin
```

2. Click **From EPMA Application**.
3. Enter the following information:
 - User name and password for the EPMA administrator
 - URL for EPMA Web Server
 - Name of the EPMA application
 - EPMA import file

4. Click **Execute**.

Convert Export Files

Convert the export file to an Oracle Data Relationship Management compatible import file. For information on converting the export file, go to [Oracle Support](#) and navigate to the white paper "How to Migrate Metadata from EPMA to DRM Application (Doc ID 2626317.1)". This white paper also contains a link to a conversion utility that you can use to convert your export file to a file that's compatible with Data Relationship Management.

For information on how to create imports for Data Relationship Management, see [Working with Imports](#).

Exporting Essbase Metadata from Performance Management Architect

Export Metadata

To export 11.1.2.4 Oracle Essbase application metadata:

1. Launch the EPMA File Generator Utility by selecting **Start, Oracle EPM Systems, Foundation Services, Performance Management Architect**, and then **Start File Generator**.

The file generator is installed with Oracle Hyperion EPM Architect by default to:

```
C:\Oracle\Middleware\EPMSys11R1\products\Foundation\BPMA  
\EPMAFileGenerator\bin
```

2. Click **From EPMA Application**.
3. Enter the following information:
 - User name and password for the EPMA administrator
 - URL for EPMA Web Server
 - Name of the EPMA application
 - EPMA import file
4. Click **Execute**.

Convert Export Files

Convert the export file to an Oracle Data Relationship Management compatible import file. For information on converting the export file, go to [Oracle Support](#) and navigate to the white paper "How to Migrate Metadata from EPMA to DRM Application (Doc ID 2626317.1)". This white paper also contains a link to a conversion utility that you can use to convert your export file to a file that's compatible with Data Relationship Management.

For information on how to create imports for Data Relationship Management, see [Working with Imports](#).

Importing Performance Management Architect Application Metadata into Data Relationship Management

You'll create a Oracle Data Relationship Management application and use the Data Relationship Management Metadata Migration Utility to load your metadata to the Data Relationship Management application where you can manage it. You must first export your metadata from Oracle Hyperion EPM Architect using the EPMA File Generator Utility and convert that file to an XML file that can be used to import the metadata to the Data Relationship Management application.

In Data Relationship Management, follow these steps to import your Performance Management Architect application metadata:

Step	Task	Refer to this documentation
1.	<ol style="list-style-type: none"> 1. Create a Data Relationship Management application to load the import file to. 2. Use the Data Relationship Management Metadata Migration Utility to load the appropriate application template, such as the Planning App Template, for the ADS type you exported using the EPMA File Generator Utility. By default, application templates are installed to: C:\Oracle\Middleware\EPMSysstem11R1\products\DataRelationshipManagement\server\apptemplates 3. Modify the Data Relationship Management configuration as necessary to align available properties in the file with the Import specification created by the application template. Create additional properties in Data Relationship Management as needed for your implementation. 	<ul style="list-style-type: none"> • Creating an Application • Migrating Data Relationship Management Metadata - Follow the instructions in "Loading Metadata" • Managing Property Definitions
2.	Import the converted file into your Data Relationship Management application.	Working with Imports
3.	Manage your metadata.	Getting Started

Exporting from Data Relationship Management and Importing to EPM Applications

Follow these steps to export your application metadata from Oracle Data Relationship Management and import it into your EPM application.

Step	Task	Refer to this documentation
1.	In Data Relationship Management: Export the application metadata to a file using the export specification created by the application template.	Working with Exports
2.	In your EPM application: Import the metadata to your application. For example, for Oracle Hyperion Planning, you can import it either interactively by dimension or via batch using the Outline Load Utility.	<ul style="list-style-type: none"> • For Planning, Importing and Exporting Data and Metadata • For Oracle Hyperion Financial Management, Managing Applications • For Oracle Hyperion Profitability and Cost Management, Importing Data into Profitability and Cost Management • For Oracle Essbase, Loading Data and Building Dimensions

3

Using Data Relationship Management with Oracle Hyperion Planning and EPM Cloud Planning

With Oracle Hyperion Planning and Oracle Enterprise Performance Management Cloud Planning, you can export and import dimensions, nodes, and properties from Oracle Data Relationship Management using a .CSV file format.

Supported dimensions are:

- Account
- Entity
- Currency
- Generic
- Attribute
- Smart Lists

Dimensions can be imported into or created in Data Relationship Management where users can then edit the properties and maintain the dimensions. This information can then be exported out of Data Relationship Management into Oracle Hyperion Planning or EPM Cloud Planning.

Additional Documentation

The following documentation will be helpful in understanding this integration.

- *Oracle Data Relationship Management Installation Guide*
- *Oracle Data Relationship Management Administrator's Guide*
- *Oracle Data Relationship Management User's Guide*
- *Oracle Enterprise Performance Management System Installation and Configuration Guide*
- *Oracle Hyperion Planning Administrator's Guide*
- *Oracle Hyperion Planning User's Guide*
- *Oracle Hyperion Planning Using Administrator Features*
- *Using Oracle Planning and Budgeting Cloud Service*
- *Administering Oracle Planning and Budgeting Cloud Service Using the Simplified Interface*

Configuring the Oracle Hyperion Planning and EPM Cloud Planning Systems

Data Relationship Management integrates with Oracle Hyperion Planning and EPM Cloud Planning using dimension files which can be exported from one system and imported into another system using standard interfaces. No additional configuration is

required in Oracle Hyperion Planning or EPM Cloud Planning to integrate with Data Relationship Management.

Configuring Planning Metadata in Data Relationship Management

See these topics:

- [Locating and Loading the Application Template](#)
- [Defining Planning Applications](#)
- [Defining and Configuring Planning Dimensions](#)
- [Dimension to Hierarchy Organization](#)
- [Configuring Planning Dimension Types](#)

Locating and Loading the Application Template

A Planning application template is installed as part of the Oracle Data Relationship Management installation. You use the template to configure the Data Relationship Management metadata to enable integration with Oracle Hyperion Planning or Oracle Enterprise Performance Management Cloud Planning.

To locate the application template:

1. On the computer where Data Relationship Management is installed, navigate to `c:\Oracle\Middleware\EPMSys11R1\products\DataRelationshipManagement\server\app-templates`.

 **Note:**

This is the default installation directory for application templates.

2. Locate the application template file named `planning-app-template.xml`.

You use the Data Relationship Management Migration Utility to load the application template. See the *Oracle Data Relationship Management Administrator's Guide* topic "Loading Metadata" for details on how to use the Migration Utility to load the application template. The application template includes Data Relationship Management metadata objects for Planning. All metadata is required for the integration to function as designed.

Defining Planning Applications

Oracle Data Relationship Management hierarchies, nodes, and properties can be loaded into a Planning application. A Data Relationship Management version can manage property values for only one application. The Planning application template includes a version-level Application property to identify the name of the application that is supported by the version.

For more information, see [Supporting Multiple Planning Applications](#).

Edit the list of values for the version-level application property for the application name to be supported.

To define the Planning application to be supported by Data Relationship Management:

1. In Data Relationship Management, from **Versions** select **Properties**.
2. From **Category**, select **Planning**, and then select the Application property for the Planning application from the category in the property grid.
3. From **Property**, select **Edit List Values**. Click **Add**.

To associate a Planning application with a version:

1. In Data Relationship Management, from **Versions** select **Properties**.
2. From **Category**, select **Planning**, and then select the Application property for the Planning application from the category in the property grid.
3. Click the **Value** cell for the property and select a value.

Defining and Configuring Planning Dimensions

Oracle Data Relationship Management hierarchies, nodes, and properties are organized into dimensions for import into Planning. The Planning application template includes a hierarchy level Dimension property. This property is used to relate and organize hierarchies by the Planning dimensions. Each hierarchy in a version can only be assigned to one dimension in the Planning application.

The Dimension property refers to hierarchy groups that are defined in the Data Relationship Management application. These properties can be populated for a hierarchy in two ways:

- Select a hierarchy group as the value for the property in the property grid.
- Assign the hierarchy to the hierarchy group using the Assign to Group menu item from the Hierarchies menu.

The list of selectable hierarchy groups for a Dimension property can be restricted to a specific subset by enabling a list of values for the property definition.

To define and configure Planning dimensions:

1. In Data Relationship Management, select a version and then select a hierarchy.
2. From **Category**, select **Planning**, and then select the Dimension property for Planning.
3. From **Property**, select **Edit List Values**.
4. Click **Add** to define dimensions to be managed in Data Relationship Management.
5. Select a Dimension property from **Group By** on the hierarchies toolbar.
6. Select a hierarchy in the version.
7. Assign the hierarchy to a hierarchy group using the **Assign to Group** menu or dragging the hierarchy.

When new dimensions are added and you do not want to use the Generic or Attribute Dimension specifications but instead use more business-related terms like Product and Ledger Code then those additional dimension types must be added as Hierarchy Groups and added to the allowed values for the Dimension property. In addition, the Dimension Type lookup property must be edited to add the lookup from the specific

business dimension to the base planning dimension type (Product to Generic or Ledger Code to Attribute).

Dimension to Hierarchy Organization

Dimensions may map to hierarchies in Oracle Data Relationship Management on a one to one or many to one basis.

A common example of one to one is the currency dimension. A common example of many to one is the entity dimension where the Top Nodes of the entity dimension are separate views of the entity by different business needs, for example Legal, Managerial, and Geographic. Splitting these branches into individual hierarchies in Data Relationship Management reduces shared nodes, is more aligned to business usage, and allows for better validations (such as all leaf nodes in Managerial must be in Legal).

The Planning integration supports both of these models. The imports are initially setup for the one to one model but can be modified by using the nodes to ignore functionality to split the incoming dimension into multiple hierarchies. The exports use hierarchy groups and in doing so support both the one to one model as well as the one to many model without changes.

Splitting the Dimension into Multiple Hierarchies in Data Relationship Management

To split the Planning dimension into multiple hierarchies the following changes to the import profile need to be made.

1. The Dimension Name needs to be added to the “Skip Nodes during Import” Section of the Filter portion of the Import Profile. For example add “Entity” to the “Skip Nodes During Import” Node list to split the entity dimension.
2. If the Dimension has members at level 2 that have no children then the “Create Base Orphan Hierarchy” option on the Style portion of the import profile needs to be selected. The following information needs to be added:
 - Name: Hierarchy Name to contain the single dimension nodes; for example Entity-Misc
 - Description: Hierarchy Name to contain the singleton dimension nodes; for example Entity Miscellaneous Branches
 - Top Node Name: Name of the top node for the hierarchy; for example Entity-Misc
 - Top Node Description: Description of the top node for the hierarchy; for example Entity Miscellaneous Branches

Configuring Planning Dimension Types

Planning dimensions defined in Oracle Data Relationship Management must have a dimension type specified in order to be imported into Planning. The Planning application template includes a hierarchy-level Dimension Type property to handle this configuration. The description of each Dimension Type property identifies the types that are supported.

In conjunction with the Dimension Type property, there is also a Dimension property that is the Hierarchy Group Property.

To configure a dimension type for Planning dimensions that have been defined:

1. From the **Planning** property category, select the Dimension Type property for Planning.
2. From **Property**, select **Edit Lookup Values** and then click **Add** and enter the dimension name in the **Key** cell and the dimension type in the **Value** cell.

The predefined dimension types lookups supported in the Planning application template are:

- Account
- Entity
- Currency
- Smart List
- Attribute
- Generic

As discussed in [Defining and Configuring Planning Dimensions](#), when new dimensions are added and you do not want to use the Generic or Attribute Dimension specifications but instead use more business-related terms like Product and Ledger Code then those must be added to the Dimension Type lookup property (Product to Generic or Ledger Code to Attribute).

Enabling Node Types

Oracle Data Relationship Management node types are automatically enabled for hierarchies that are imported from Oracle Hyperion Planning or Oracle Enterprise Performance Management Cloud Planning. New hierarchies that are created directly in Data Relationship Management will not have node types enabled without some additional configuration. Perform one of the following actions to enable node types for new hierarchies to be exported to Oracle Hyperion Planning or EPM Cloud Planning:

- Assign HP.MemberType as the default value of the Hierarchy Node Type property definition in the Core namespace. This configuration will enable node types for all hierarchies using the Member Type property in the HP namespace. This task can be performed only by an application administrator.
- Select the Member Type property from the HP namespace for the value of the Hierarchy Node Type property in the System category for each new hierarchy created.

Configuring Oracle Hyperion Planning Export Profiles

Export profiles are included in the application template to output dimensions to Oracle Hyperion Planning. The export profiles are:

- Planning Account
- Planning Attribute
- Planning Currency
- Planning Entity
- Planning Generic

- Planning SmartList

These export profiles use the Dimension and Dimension Type properties configured for versions in Data Relationship Management to determine which hierarchies, nodes, and properties should be exported to Oracle Hyperion Planning. The exports are configured to use the Membership properties managed for hierarchy nodes to filter the export results to only nodes that are marked as Parent Members or Base Members.

You can add a filter to an export profile, add additional columns, and make other custom modifications to the export profiles loaded from the Oracle Hyperion Planning application template. You will need to modify these exports to support optional configurations for multiple aliases, different Plan Types, Weeks Distribution, Attribute Dimensions, and other Oracle Hyperion Planning-specific capabilities.

Two key points when modifying the export profile are:

- Ensure that the properties needed for Oracle Hyperion Planning are present in the columns section of the export.
- Ensure that the header record matches the columns in the export and specifies the correct label for the outline load utility for that column.

It is best practice to make a copy of the predefined export profiles and use the copy to modify/ customize so that you have easy access back to the original if needed. If the original is lost or deleted it can be reloaded from the application template.

Optimal Configurations

For more information see:

- [Multiple Aliases](#)
- [Multiple UDAs](#)
- [Attribute Dimensions](#)
- [Generic and Custom Dimensions](#)
- [Supporting Multiple Planning Applications](#)
- [Plan Types](#)

Multiple Aliases

The Oracle Hyperion Planning application template is configured with a single Alias property for each type. In some cases, multiple aliases may be necessary to maintain dimension member descriptions in different languages.

To support multiple aliases:

1. Create a new custom global node Alias property for each additional description to be supported.
2. Do one of the following:
 - Create a new custom MemberAliasLength validation for each additional custom Alias property.
 - Modify the existing MemberAliasLength validation to check the additional alias as well as the default alias.
3. Assign the custom Alias properties and MemberAliasLength validations to the node types for the Oracle Hyperion Planning node types.

4. Assign any new custom MemberAliasLength validations to versions and hierarchies where required.
5. Modify Oracle Hyperion Planning type exports for dimensions where custom Aliases are required.
 - a. Add new custom Alias properties to the columns of the export.
 - b. Add the labels for the new Alias properties to the header record of the export.
 - c. Assign new custom Alias validations.

Multiple UDAs

Multiple UDAs for a node in Oracle Data Relationship Management are supported by entering them in a comma-delimited format in the appropriate UDA property.

Attribute Dimensions

Oracle Hyperion Planning offers the ability to use custom Attribute dimensions. The Attribute dimension type is supported by the Oracle Hyperion Planning application template, however, some additional configuration is necessary in order to account for the user-defined aspects for supporting them.

To configure Attribute dimensions in Data Relationship Management:

1. Create a new custom global node property for each Attribute dimension that will manage the association of a base dimension member to a custom attribute dimension member.
2. Create a new custom version property of data type ListGroup to use for association of base dimension members to members of the Attribute dimension. This property will identify the valid hierarchies from which an attribute member can be selected. Go to the Version and set up the list of allowed hierarchies for the attribute dimension.
3. Create a new custom formula validation for each Attribute dimension to enforce dimension associations between it and base dimension members that use the Attribute dimension. Compare the selected values in the custom version ListGroup property for the Attribute dimension to the value of the Core.References property for the Attribute dimension member being referenced.

For example, the Global node attribute property is Custom.Attribute and the version-level attribute hierarchies property is Custom.AttributeHiers. Invalid attribute member validation formula:

```
Or(IsBlank(PropValue(Custom.Attribute)),GreaterThanOrEqual(ArrayCount(Intersection(NodePropValue(PropValue(Custom.Attribute)),Custom.AttributeHiers,NodePropValue(PropValue(Custom.Attribute)),Core.References),[comma]),[comma]),1))
```

4. Assign the custom global node Attribute properties and validations to the appropriate node types for the dimensions to be supported.
5. Assign any new custom Attribute validations to versions and hierarchies where required.
6. Modify Oracle Hyperion Planning exports for the attribute dimension and for the base dimension mapped to it.

- a. Create the export profile for the new attribute dimension. Copy the predefined attribute dimension for a start and customize as needed for the Oracle Hyperion Planning configuration (Aliases, Plan Types and so on).
 - b. Assign new custom Attribute validations to the base dimension export.
 - c. Add custom Attribute properties as exports columns to the base dimension export and update the export header to match.
7. Add the new Attribute Dimensions to the Attribute Hierarchy Group and set the Hierarchy Dimension Property to Attribute.

For more information on customizing the new attribute dimensions, see [Defining and Configuring Planning Dimensions](#) and [Configuring Planning Dimension Types](#).

Generic and Custom Dimensions

Oracle Hyperion Planning supports additional Generic dimensions. Many clients use them for dimensions like Product. The Oracle Hyperion Planning application template supports the Generic Dimensions.

To configure a new generic dimensions in Data Relationship Management:

1. Create the new hierarchies for the Generic dimension.
2. Create the export profile for the new generic dimension. Copy the predefined generic dimension for a start and customize as needed for the Oracle Hyperion Planning configuration (Aliases, Plan Types, and so on).
3. Add the new Generic hierarchies to the Generic Hierarchy Group and set the Hierarchy Dimension Property to Generic.

For more information on customizing the new attribute dimensions, see [Defining and Configuring Planning Dimensions](#) and [Configuring Planning Dimension Types](#).

Supporting Multiple Planning Applications

Supporting Production and Testing Instances of Oracle Hyperion Planning

There are no additional configuration steps required in Data Relationship Management to support testing and production Oracle Hyperion Planning applications. The same exports can be used to generate the outline load utility files that can be loaded into a test application and then later a production application.

Supporting Multiple Different Oracle Hyperion Planning Applications

There are two methods that can be used to support multiple different Oracle Hyperion Planning applications.

1. Separate Data Relationship Management Versions
 - Each Oracle Hyperion Planning instance has its dimensional data in a different version.
 - Additional properties are not required.
 - Hierarchies cannot be shared across versions so each would be independent of the other.
 - Exports may need to be made specific to each Oracle Hyperion Planning instance if there are differences in Oracle Hyperion Planning configurations that require a different column set (Plan Types, Aliases, Weeks Distribution setup, Attribute & Generic Dimensions).

2. Custom Data Relationship Management Properties and Exports

- Each additional instance of an Oracle Hyperion Planning application requires separate custom Dimension and Dimension Type properties if the application instances have different dimensions.
- Each additional instance of an application requires separate custom Membership, Member, and Parent properties if different dimension members are required for each application instance.
- Each additional instance of an application requires separate Oracle Hyperion Planning properties if these need to be different by the application instance (For example UDA, Data Storage and so on).
- Each additional instance of an application requires separate custom Oracle Hyperion Planning dimension exports if there are differences in Oracle Hyperion Planning configurations that require a different column set (Plan Types, Aliases, Weeks Distribution setup, Attribute, and Generic Dimensions) or if there are different Planning properties being used.

Plan Types

Initial Predefined Plan Types

The Planning application integration is designed with the initial three default Plan Types defined in the properties and in the export and import profiles:

- Plan1
- Plan2
- Plan3

Additional Predefined Plan Types

The following additional Plan Types are predefined for ease of use and have the properties needed to support them but are not included in the default import and export profiles:

- Capex
- Hcp
- Project
- WorkForce

Plan Type Properties

For each Plan Type (Plan1, Plan2, Plan3, Capex, Hcp, Project, and Workforce) the following properties are defined:

- Aggregation – Plan Specific Aggregation
Local Node Defined String Property with Allowed Values (Default: Add)
- Aggregation Code – Plan Specific Aggregation Code
Local Node Lookup String Property using Aggregation as the lookup
- Data Storage – Plan Specific Data Storage
Local Node Formula Derived String Property, Overridable– Returns the non plan-specific Data Storage (HP.DataStorage) unless overridden.
- Formula – Plan Specific Formula

Global Node Formula Derived Formatted Memo Property, Overridable—Returns the non plan specific Formula (HP.Formula) unless overridden.

- Valid For Plan – Indicates if the member is valid for the plan.

Local Node Defined Boolean Property (Default: TRUE)

Enabling Additional Predefined Plan Types

The additional plan types for Capex, Hcp, and Workforce can be enabled by adding the properties to the import and export profiles. For the import profiles the properties for the plan type need to be added to the appropriate location in the columns of the relationship section of the import. For the export profiles the properties need to be added to the export columns and the export header must be modified to add the labels needed for the Outline Load Utility.

Adding New Plan Types

When a new plan type is to be added to the Planning system it must also be configured in Data Relationship Management. Custom properties need to be added for the new plan type and can be modeled after the properties for the predefined plan types:

- Aggregation – Plan Specific Aggregation
- Aggregation Code – Plan Specific Aggregation Code
- Data Storage – Plan Specific Data Storage
- Formula – Plan Specific Formula
- Valid For Plan – Indicates if the member is valid for the plan.

In addition, the import and export profiles must be modified to add the new properties to the column sections as discussed in **Enabling Additional Predefined Plan Types**. New plan types must be added to the HP.SourcePlanType list of allowed values.

Plan Types Differing by Dimension

Some dimensions can be set up in Planning to only be in specific plan types. The import and export profiles must be modified to either add the plan types to the appropriate import and export profiles or to remove them from the appropriate import and export profiles to match the Planning system.

Weekly Distribution Setup

The Planning application template provides the support for the Weeks Distribution functionality. If the Weeks Distribution capability of Planning is being used, then you need to perform the following set up on the Data Relationship Management application:

- Set the HP.WeeksDistributionApp property to the appropriate value for the Planning application.
- Set the HP.WeeksDistribution property to True on the nodes that will use the Weeks Distribution.
- Include the HP.WeeksDistribution property in columns for the appropriate dimension exports and modify the header record for the export to include the label for the column as required by the outline load utility.

 **Note:**

The export header record label can always be “Use 445” even if other weekly distributions are being used.

Initializing Data Relationship Management with Planning Dimensions

Exporting Dimension Members from Oracle Hyperion Planning

To populate Oracle Data Relationship Management from an existing Oracle Hyperion Planning application the dimensions can be exported using the Web interface or the Outline Load Utility formatted dimension file (one per dimension) so that they can be imported into Data Relationship Management.

See “Loading Data and Metadata” in the *Oracle Hyperion Planning Administrator's Guide* for more information on using the Outline Load Utility for Planning.

See [Importing Planning Dimensions into Data Relationship Management](#) for details on the import process into Data Relationship Management.

Planning Outline Load Utility Command Line

To process the outline load utility dimension files into Data Relationship Management:

1. Create the Outline Load Utility files for each dimension from the Oracle Hyperion Planning application.
2. Deliver the dimension files to a location that can be accessed by Data Relationship Management.
3. Run the import profiles in Data Relationship Management for each dimension.

To export the dimensions using the Outline Load Utility command line, run the OutlineLoad command for each dimension with the following options:

- [/M] - Headers
- Specify the dimension by one of the following:
 - [/D:loadDimensionName]–Dimension to Process
 - [/DU:userDefinedLoadDimensionName]–User Defined Dimension to Process
 - [/DA:attributeLoadDimensionName:baseDimensionName]–Attribute/Base Dimension to Process
 - [/DAT:attributeLoadDimensionName:baseDimensionName]–Text Attribute/ Base Dimension to Process
 - [/DAN:attributeLoadDimensionName:baseDimensionName]–Numeric Attribute/ Base Dimension to Process
 - [/DAB:attributeLoadDimensionName:baseDimensionName]–Boolean Attribute/ Base Dimension to Process
 - [/DAD:attributeLoadDimensionName:baseDimensionName]–Date Attribute/ Base Dimension to Process

- [/DS:HSP_SMARTLISTS]–Smart Lists Dimension to Process
- [/E:outputFileName]–Output Filename
- [/8]–Use UTF8 File (Default)

For example: `OutlineLoad -f:D:\Files\PswdFile.txt /A:Forecast /U:admin /M /D:Entity /E:D:\Files\Extracts\EntityMbrs.txt /L:D:\Logs\EntityMbrs.log /X:D:\Logs\EntityMbrs_Excptns.log`

Exporting Dimension Members from EPM Cloud Planning

To populate Data Relationship Management from an existing Oracle Enterprise Performance Management Cloud Planning application the dimensions can be exported using the Web interface or the EPM Automate Utility which provides the ability to remotely perform the export tasks.

For more information, see "Using the EPM Automate Utility" in "Using Oracle Planning and Budgeting Cloud Service" and "Importing and Exporting into a Planning Application" in "Administering Planning for Oracle Planning and Budgeting Cloud Service".

See "Exporting Metadata" in "Administering Oracle Planning and Budgeting Cloud Service Using the Simplified Interface".



Note:

EPM Cloud Planning jobs must be configured for each of the tasks that are to be performed with the EPM Automate Utility. For the Data Relationship Management integration this includes any metadata import and export tasks that are to be run with EPM Automate. See "Scheduling Jobs" in "Administering Oracle Planning and Budgeting Cloud Service Using the Simplified Interface".

To export metadata from EPM Cloud Planning using EPM Automate:

1. Login to EPM Cloud Planning.
Syntax: `epmautomate login username password url identitydomain`
2. Execute the EPM Cloud Planning export metadata job (can include one or more dimensions).
Syntax: `epmautomate exportmetadata jobname filename`
3. Download the metadata file. It will be in a zipped format.
Syntax: `epmautomate downloadfile filename`
4. Log out of EPM Cloud Planning.
Syntax: `epmautomate logout`

Importing Planning Dimensions into Data Relationship Management

After Data Relationship Management has been configured to match the Planning systems (Aliases, Plan Types, Weeks Distribution, Attribute Dimensions, and Generic/Custom Dimensions) then the data from the Planning systems can be imported into Data Relationship Management. This is designed to allow initial population of the Data

Relationship Management hierarchies from existing Planning applications. It is not intended to be used as an ongoing process.

 **Note:**

You do not need to do the import if the dimensions already exist or will be created in Data Relationship Management.

The imports are setup to do the following:

- Process files in UTF-8 encoding
- Assign hierarchy-level validations
- Set the dimension property
- Define the sort order property based on the record order
- Determine leafs at the end of the import process
- Warn on shared nodes (not error)

After running the imports the dimensions will be in separate versions. The Blender functionality can be used to combine the dimensions into a single version.

To import Planning dimensions into Data Relationship Management:

1. Unzip the files to a location accessible to Data Relationship Management for import.
2. You can use the Data Relationship Management Web Client or Batch Client to import Planning dimensions and blend them.
 - a. Run the Data Relationship Management import for each Dimension file.
 - b. Run a Data Relationship Management Blender to combine the dimensions into the production version.

Managing Planning Dimension Members in Data Relationship Management

Planning dimension members are managed in Oracle Data Relationship Management using nodes, hierarchies, and properties. Each node represents a single dimension member. Nodes may be organized into hierarchies which can be used in one or more Planning dimensions. The Planning application template for Data Relationship Management includes node-level dimension member properties for Planning and membership properties used to mark hierarchy nodes as dimension members. Nodes are validated as dimension members for Planning using validations from the Planning application template. After all validations have successfully run, this information can be imported into Planning using Dimension exports that create files that can be processed by Planning Outline Load Utility. In addition, the Planning application template provides import profiles for importing the dimension from Planning into Data Relationship Management.

Managing Planning Dimension Membership

Membership properties are used to flag nodes in Oracle Data Relationship Management that should be managed and exported as Planning dimension members. The membership property can be used to include only a subset of nodes in a Data Relationship Management hierarchy as members of a Planning dimension.

For hierarchies to be included in a Planning dimension, the top-level node or nodes must be updated to define the Planning membership (HP.Membership). By default, this resolves to Not Applicable. All nodes that have a membership of Not Applicable are filtered from the exports that produce the outline load utility files to feed the Planning systems.

The Membership property has the following allowed values:

- Not Applicable
- Dimension
- Parent Member
- Base Member

For hierarchies that have a top node that is the same as the dimension, the top node membership property should be set to Dimension. For hierarchies where the top node or nodes are dimension members, set the membership to Parent Member.

In addition, if the Sort Order is desired then the Hierarchy Sort Order Property for the Hierarchy needs to be configured to point to the Sort Order property.

It is best practice to make a copy of the predefined import profiles and use the copy to modify so that you have easy access back to the original if needed. If the original is lost or deleted, it can be reloaded from the application template.

To assign Planning dimension membership:

1. Open a hierarchy that has been assigned to a Planning dimension.
2. Select a node in the hierarchy, and from **Category**, select **Planning**.
3. Select the Planning Membership property.
4. Do one of the following:
 - To mark the node and its descendants as members of the Planning dimension, set the Membership property value to Parent Member.
 - To mark a node with children as a bottom-level member of the Planning dimension, set the Membership property value to Base Member.

 **Note:**

Setting node membership to Base Member causes the nodes below it to become Not Applicable. This allows you to filter out the bottom section of a branch from a Planning application.

Automatic Membership to Planning Dimensions

If you want all imported or newly created hierarchies to automatically have membership to a Planning dimension, then the following formula for HP.Membership can be used.

```
If (Equals (String, PropValue (HP.Dimension) , ) , Not
Applicable , If (Equals (Integer, PropValue (Core.Level) ,
1) , If (Equals (String, PropValue (HP.Dimension) , PropValue (Core.PrimaryName_
MDM) ) , Dimension , Parent
Member) , If (Or (Equals (String, PropValue (HP.ParentMembership) , Dimension) , E
quals (String, PropValue (HP.ParentMembership) , Parent
Member)) , If (PropValue (Core.Leaf) , Base Member , Parent Member) , Not
Applicable)))
```

Managing Planning Dimension Member Properties

Nodes in hierarchies that are associated with a Planning dimension have dimension member properties. These properties correspond to the same dimension member properties available in Planning that are used by the outline load utility for importing and exporting dimension metadata from Planning.

Dimension member property values should be defined appropriately for each node that is included in a hierarchy associated with a Planning dimension using a hierarchy-level Dimension property and marked as a member of the dimension using a node-level Membership property.

See the "Dimension Properties" section in the Loading Data and Metadata chapter of the *Oracle Hyperion Planning Administrator's Guide* for more information on the use of these properties in Planning. See the Planning application template report for a complete list of properties available in Data Relationship Management for Planning dimension members.

Validating Planning Dimension Members and Properties

Oracle Data Relationship Management hierarchy nodes can be validated as Planning dimension members before being exported to Planning. These validations can be run for a version, hierarchy, or node.

The validations loaded from the Planning application template must be assigned to versions, hierarchies, or nodes in order to run as real-time validations. If the hierarchies are imported using the imports provided in the application template then the hierarchy-level validations will be assigned. These validations are assigned to the exports from the Planning application template to run as batch validations prior to being exported to Planning.

Refer to the Planning application template report for a complete list of validations available in Data Relationship Management for Planning dimension members

Exporting Dimension Members from Data Relationship Management

Use these steps to export dimension files from Oracle Data Relationship Management and import the metadata into the Oracle Enterprise Performance Management Cloud Planning.

 **Note:**

EPM Cloud Planning jobs must be configured for each of the tasks that are to be performed with the EPM Automate Utility. For the Data Relationship Management integration this includes any metadata import and export tasks that are to be run with EPM Automate. See "Scheduling Jobs" in "Administering Oracle Planning and Budgeting Cloud Service Using the Simplified Interface".

To export Planning dimensions from Data Relationship Management, use the Data Relationship Management Batch Client to export dimension files in .CSV format.

Importing Dimension Members into Oracle Hyperion Planning

Hierarchies, nodes, and properties managed in Oracle Data Relationship Management are loaded as dimension members using the Web interface or the Outline Load Utility in Oracle Hyperion Planning. Outline Load Utility files are created with Data Relationship Management export profiles (one per dimension) and a version or version variable.

See "Loading Data and Metadata" in the *Oracle Hyperion Planning Administrator's Guide* for more information on using the Outline Load Utility for Oracle Hyperion Planning.

To process the Outline Load Utility dimension files into Oracle Hyperion Planning:

1. Deliver the files to a location where the Oracle Hyperion Planning application can process them.
2. Run the outline load utility on the Oracle Hyperion Planning application to process the dimension files.

Oracle Hyperion Planning Outline Load Utility Command Line

To import the dimensions using the Outline Load Utility command line, run the OutlineLoad command for each dimension with the following options:

- [/M] - Headers
- Specify the dimension by one of the following:
 - [/D:loadDimensionName]–Dimension to Process
 - [/DU:userDefinedLoadDimensionName]–User Defined Dimension to Process
 - [/DA:attributeLoadDimensionName:baseDimensionName]–Attribute/Base Dimension to Process
 - [/DAT:attributeLoadDimensionName:baseDimensionName]–Text Attribute/ Base Dimension to Process
 - [/DAN:attributeLoadDimensionName:baseDimensionName]–Numeric Attribute/ Base Dimension to Process
 - [/DAB:attributeLoadDimensionName:baseDimensionName]–Boolean Attribute/ Base Dimension to Process

- [/DAD:attributeLoadDimensionName:baseDimensionName]–Date Attribute/
Base Dimension to Process
- [/DS:HSP_SMARTLISTS]–Smart Lists Dimension to Process
- [/E:outputFileName]–Output Filename
- [/8]–Use UTF8 File (Default)

For example: OutlineLoad -f:D:\Files\PswdFile.txt /A:Forecast /U:admin /M /
D:Entity /E:D:\Files\Extracts\EntityMbrs.txt /L:D:\Logs\EntityMbrs.log /
X:D:\Logs\EntityMbrs_Excptns.log

Importing Dimension Members into EPM Cloud Planning

Hierarchies, nodes, and properties managed in Oracle Data Relationship Management are loaded as dimension members using the Web interface or the EPM Automate Utility in which provides the ability to remotely perform the import tasks.

To import dimensions into Oracle Enterprise Performance Management Cloud Planning using EPM Automate:

1. Upload the dimension files created by the Data Relationship Management export to EPM Cloud Planning.

Syntax: epmautomate uploadfile filename

2. Execute the import metadata job.

Syntax: epmautomate Importmetadata jobname filename

For more information, see:

- "Using the EPM Automate Utility" in "Using Oracle Planning and Budgeting Cloud Service"
- "Importing and Exporting into a Planning Application" in "Administering Planning for Oracle Planning and Budgeting Cloud Service"
- "Loading the Import File" in "Administering Oracle Planning and Budgeting Cloud Service Using the Simplified Interface"

4

Using Data Relationship Management with FDMEE

Oracle Data Relationship Management may be used to manage financial master data entities which include chart of account values and hierarchies used for general ledgers in ERP systems such as E-Business Suite (EBS), Peoplesoft (PS), and Fusion Financials(FS). Data Relationship Management is also used to manage dimension members and hierarchies for EPM applications such as Financial Management, Planning, and Essbase, which often source transactional data from these same ERP systems. Oracle Hyperion Financial Data Quality Management, Enterprise Edition is used to load transactional data from these ERP systems to Oracle EPM applications.

In some cases, the dimensionality and granularity of the target EPM application does not match the source system and mappings must be defined between the source and target in order to appropriately transfer and load the transactional data from the ERP system to the EPM application. FDMEE provides the ability to natively manage mappings between source ERP segment values and target EPM dimension members explicitly or using rules (range, wildcard, script). When Data Relationship Management is used as a master for dimension members of the EPM applications, Data Relationship Management may also be used to manage mappings of ERP segment values to those dimension members and synchronize those changes to FDMEE for use by transactional data loading processes.

The main advantage of maintaining explicit mappings in Data Relationship Management is to be able to validate integrity and record auditing information for each mapping at the time of making the change. This is not easily accomplished with rule-based mappings in FDMEE. You can use ancestor relationships from Data Relationship Management hierarchies to map multiple values at the same time or you can use queries/compares with multi-select to map multiple values at the same time.

For each FDMEE location, the user gets to decide whether to use explicit mappings or rule based mappings. If rule-based mappings are preferable, then use FDMEE for those mappings. If you want to use Data Relationship Management to manage the mappings along with the GL segment values and/or EPM dimension members being managed in Data Relationship Management, then you would use explicit mappings in Data Relationship Management and load those mappings into FDMEE.

Integration

The 11.1.2.4.340 release of Oracle Data Relationship Management provides an integration with the 11.1.2.4.200 release of Oracle Hyperion Financial Data Quality Management, Enterprise Edition.

The integration with FDMEE provides the following capabilities:

- Import chart of accounts values and hierarchies from source ERP systems to Data Relationship Management
- Map ERP source values to EPM target dimension members within Data Relationship Management

- Export ERP source to EPM target member mappings from Data Relationship Management to FDMEE

Prerequisites

The Oracle Hyperion Financial Data Quality Management, Enterprise Edition application template references property definitions in the Performance Management Architect and Planning application templates. One of these templates must be loaded into Oracle Data Relationship Management before the FDMEE template is loaded.

In addition to the Performance Management Architect and Planning application templates, this integration requires valid Performance Management Architect or Planning target system dimensions to be set up for the mapping to be able to be done and exported back to FDMEE.

There are two main use cases:

- Existing Performance Management Architect or Planning system dimensions to be mapped to
- Creation of new Performance Management Architect or Planning dimensions from the FDMEE source system imports to be mapped to

Additional work will be required to use this application template for custom Data Relationship Management implementations that are not using the Performance Management Architect or Planning templates. Details on using this application template with a custom Data Relationship Management implementation are detailed in the “Configuring the FDMEE Template for Custom Data Relationship Management Implementations” section.

Additional Documentation

The following documentation will be helpful in understanding the integration of Oracle Data Relationship Management with Oracle Hyperion Financial Data Quality Management, Enterprise Edition.

- *Oracle Data Relationship Management Installation Guide*
- *Oracle Data Relationship Management Administrator's Guide*
- *Oracle Data Relationship Management User's Guide*
- *Oracle Hyperion Financial Data Quality Management Administrator's Guide*

Configuring the FDMEE System Environment

For information on configuring the Oracle Hyperion Financial Data Quality Management, Enterprise Edition environment, see “Integrating Data Relationship Management with FDMEE” in *Oracle Hyperion Financial Data Quality Management Administrator's Guide*.

Installing and Configuring Foundation Services

You must install and configure Oracle Hyperion Foundation Services for Oracle Data Relationship Management. See the *Oracle Enterprise Performance Management System Installation and Configuration Guide*.

Deploying and Configuring the Data Relationship Management Web Service

FDMEE uses the Data Relationship Management Web service to export dimension members, properties, and relationships for import into the Shared Library or an individual EPM application. This Web service must be deployed and configured to enable this integration.

See “Deploying and Configuring the Data Relationship Management Web Service API” in the *Oracle Data Relationship Management Installation Guide*.

Configuring FDMEE Metadata in Data Relationship Management

Related Topics

- [FDMEE Application Template](#)
- [Planning Application Template Metadata](#)
- [Locating and Loading the Application Template](#)
- [Configuring the External Connections to the FDMEE Database](#)
- [Reconnecting Imports and Exports to the FDMEE Database](#)
- [Configuring Domains](#)
- [Configuring Blend After Import](#)
- [Extending Imports to Use Additional Custom Top Member and Aggregation](#)
- [Adding Mapping Validations](#)

FDMEE Application Template

The Oracle Hyperion Financial Data Quality Management, Enterprise Edition Application template provides the following metadata for Oracle Data Relationship Management that supports the FDMEE integration.

Property Definitions {Full Qualified Name (Label)}

- EPMA.MemberType(Member Type)
- FDMEE.Description_Brazilian_Portuguese (Description Brazilian Portuguese)
- FDMEE.Description_French (Description French)
- FDMEE.Description_German (Description German)
FDMEE.Description_Japanese (Description Japanese)
FDMEE.Description_Korean (Description Korean)
- FDMEE.Description_Latin_American_Spanish (Description Latin American Spanish)
- FDMEE.Description_Portuguese (Description Portuguese)
- FDMEE.Description_Spanish (Description Spanish)
- FDMEE.Description_Traditional_Chinese (Description_Traditional_Chinese)
- FDMEE.Dimension (Dimension)

- FDMEE.EPM_App (EPM App)
- FDMEE.EPM_Location_Keys (EPM Location Keys)
- FDMEE.EPM_Location_Names (EPM Location Names)
- FDMEE.ERP_Acct_Entity (ERP Acct Entity)
- FDMEE.ERP_Segment (ERP Segment)
- FDMEE.ERP_Src_Name (ERP Src Name)
- FDMEE.Hierarchy_Dimension (Hierarchy Dimension)
- FDMEE.Hierarchy_Domain_Qualifier (Hierarchy Domain Qualifier)
- FDMEE.Hierarchy_ID (Hierarchy ID)
- FDMEE.Hierarchy_Table_Name (Hierarchy Table Name)
- FDMEE.Load_ID (Load ID)
- FDMEE.Member_Table_Name (Member Table Name)
- FDMEE.Node_Domain_Qualifier (Node Domain Qualifier)
- FDMEE.Node_Type (Node Type)
- FDMEE.SRCKEY_Name (SRCKEY Name)
- FDMEE.Version_EPM_Location_Keys (Version EPM Location Keys)
- FDMEE.Version_EPM_Location_Names (Version EPM Location Names)
- FDMEE.Version_Eff_Date (Version Eff Date)

Property Category

FDMEE

Node Type

FDMEE ERP—Node type for imported ERP nodes with FDMEE properties

Import

- FDMEE Import—Generic import with no target application values
- FDMEE HFM Import—Import with HFM target values
- FDMEE Planning Import—Import with Planning target values
- FDMEE Essbase(ASO) Import—Import with Essbase ASO target values
- FDMEE Essbase(BSO) Import—Import with Essbase BSO target values

Export

- FDMEE HFM Map Export—Mapping export using HFM names and dimension types
- FDMEE Planning Map Export—Mapping export using Planning names and dimension types
- FDMEE Planning Map Export—Mapping export using Essbase names and dimension types

Blender

FDMEE Blend—Blender to merge imported FDMEE dimensions into an existing version

Version Variable

FDMEE_MAP_VERSION—Version variable used by the exports to FDMEE

External Connection

- FDMEE Import DB—External Connection for imports
- FDMEE Export DB—External Connection for exports

System Preferences

InvDescr—System preference for invalid description characters

Planning Application Template Metadata

The following property definitions are used from the Oracle Hyperion Planning application template.

Property Definition	Property Definition
HP.AccountType (Account Type)	HP.ParentName (HP Parent)
HP.Capex.ValidForPlan (Valid For Plan (Capex))	HP.Plan1.ValidForPlan (Valid For Plan (Plan 1))
HP.DataStorage (Data Storage)	HP.Plan2.ValidForPlan (Valid For Plan (Plan 2))
HP.Dimension (HP Dimension)	HP.Plan3.ValidForPlan (Valid For Plan (Plan 3))
HP.DimensionType (HP Dimension Type)	HP.SourcePlanType (Source Plan Type)
HP.MemberName (HP Member)	HP.TimeBalance (Time Balance)
HP.MemberType (Member Type)	HP.VarianceReporting (Variance Reporting)
HP.Membership (HP Membership)	HP.Wrk.ValidForPlan (Valid For Plan (Workforce))
HP.ParentMembership (HP Parent Membership)	

Locating and Loading the Application Template

The Oracle Hyperion Financial Data Quality Management, Enterprise Edition application template is installed as part of the Oracle Data Relationship Management installation. You use the template to configure the Data Relationship Management metadata to enable integration with FDMEE.

To locate the FDMEE application template:

1. On the computer where Data Relationship Management is installed, navigate to c:\Oracle\Middleware\EPMSys11R1\products\DataRelationshipManagement\server\app-templates.

 **Note:**

This is the default installation directory for application templates.

2. Locate the application template file named `fdmee-app-template.xml`.

You use the Data Relationship Management Migration Utility to load the application template. See the *Oracle Data Relationship Management Administrator's Guide* topic "Loading Metadata" for details on how to use the Migration Utility to load the application template.

The FDMEE application template includes Data Relationship Management metadata objects for FDMEE and references to property definitions in the Planning application template. During the template load process, you can select all metadata objects from the template or select only objects specific to a particular EPM application type. For example, you might select property definitions, imports, and exports specific to Planning.

The property category, node type, and blender included in the FDMEE template reference properties of multiple application types. In situations where you are using the FDMEE template with a single EPM application type, the template load process will identify missing dependencies to properties of other application types. Select the "Ignore Dependencies" option in the Migration Utility to load these metadata objects without these references.

Possible Conflicts to Review Before Loading the Application Template

This application template contains a modification to the `EPMA.MemberType` formula that adds the FDMEE ERP Node Type for ERP nodes from FDMEE. If you have modified the original formula for the `EPMA.MemberType` property in the Performance Management Architect or Planning templates, then your changes will be overridden unless you deselect this property during the migration utility load process. The View Differences function of the Data Relationship Management Migration Utility can be used to compare the difference of this property before loading. If the formula has been customized, then it is recommended that you not load the value from the FDMEE application template. The formula will need to be manually updated to include the logic for the FDMEE ERP node types.

This application template contains two system preferences (`InvName` and `InvDescr`). Before loading the application template, review the values for these preferences against the values currently set in your system. This can be done manually or with the View Differences function of the Data Relationship Management Migration Utility. You do not have to migrate these preferences especially if your configuration is more conservative for business reasons. The import is set up to not enforce invalid node name characters so that the FDMEE nodes can be imported even if they have invalid characters in them.

Supported Languages

The FDMEE application template includes support for 10 languages for node descriptions loaded into Data Relationship Management during a FDMEE metadata load process. Supported languages are:

- Brazilian Portuguese
- English
- French

- German
- Japanese
- Korean
- Latin American Spanish
- Portuguese
- Spanish
- Traditional Chinese

Configuring the External Connections to the FDMEE Database

The external connections must be properly configured before viewing, running, or editing the import and export defined in the application template.

Oracle Hyperion Financial Data Quality Management, Enterprise Edition writes to these tables:

- AIF_HS_DRM_LOADS—Version import section
- AIF_HS_DRM_LOAD_HIERARCHIES—Hierarchy import section
- AIF_HS_DRM_MEMBER_V—Node import section
- AIF_HS_DRM_HIERARCHY_V—Relationship import section

Oracle Data Relationship Management exports the member mappings to the TDATAMAP_STG table.

Note:

The FDMEE database must be in place and set up before performing this task.

After loading the application template, follow these procedures to configure the external connections:

- [FDMEE Import Database](#)
- [FDMEE Export Database](#)

FDMEE Import Database

To configure the FDMEE import database:

1. Login to the Data Relationship Management system with an account that has the administrator role.
2. On the Home page, select **Administer**.
3. Expand **External Connections** and double click the **FDMEE Export DB** connection.
4. Define the Connection String, User ID and Password to connect to the FDMEE database.

5. Click **Test Connection** to verify that a connection is made with the credential entered.
6. Select the **Include Views** checkbox.
7. Fill in the Filter (Schema and Object) with the schema for the FDMEE database and AIF_HS_DRM* and then click **Retrieve Objects**.
8. Remove the initial FDMEE_INTEGRATION.AIF_HS_DRM* tables (4) by clicking **Remove All**.
9. Highlight the four tables that start with AIF_HS_DRM* in the **Available** objects and then click **Select** to move them to the **Selected** column.
10. Click **Save** to save the changes to the external connection.

FDMEE Export Database

To configure the FDMEE export database:

1. Login to the Data Relationship Management system with an account that has the administrator role.
2. On the Home page, select **Administer**.
3. Expand **External Connections** and double click the **FDMEE Export DB** connection.
4. Define the Connection String, User ID and Password to connect to the FDMEE database.
5. Click **Test Connection** to verify that a connection is made with the credential entered.
6. Fill in the Filter (Schema and Object) with the schema for the FDMEE database and TDATAMAP_STG and then click **Retrieve Objects**.
7. Remove the initial FDMEE_INTEGRATION.TDATAMAP_STG by clicking **Remove All**.
8. Highlight TDATAMAP_STG in the **Available** objects and then click **Select** to move it to the **Selected** column.
9. Click **Save** to save the changes to the external connection.

Reconnecting Imports and Exports to the FDMEE Database

After configuring the external connections, the exports and imports must be reconnected to the Oracle Hyperion Financial Data Quality Management, Enterprise Edition database tables (this is due to the schema change made in the external connection).

Note:

The Oracle Data Relationship Management user needs to have the Application Administrator role.

To reset exports:

1. On the Data Relationship Management Home page, click **Export**.
2. Perform steps 3-8 for each of these exports:
 - FDMEE Essbase Map Export
 - FDMEE HFM Map Export
 - FDMEE Planning Map Export
3. Open the export by double clicking it.
4. Set a version and top node. These can be changed later, but are required to save the changes.
5. Click on the **Target** tab.
6. Set the **Database Table** drop down to the TDATA_MAP_STG table.
7. Verify that all of the warnings in the **Database Column Options** are gone.
8. Click **Save**.

To reset imports:

1. On the Data Relationship Management page, click **Import**.
2. Perform steps 3-8 for each of these imports:
 - FDMEE Essbase(ASO) Import
 - FDMEE Essbase(BSO) Import
 - FDMEE HFM Import
 - FDMEE Planning Import
 - FDMEE Import
3. Open the import by double clicking it.
4. Set the **Version Source Object** to the AIF_HS_DRM_LOADS table.
5. Set the **Hierarchy Source Object** to the AIF_HS_DRM_LOAD_HIERARCHIES table.
6. Set the **Node Source Object** to the AIF_HS_DRM_MEMBER_V view.
7. Set the **Relationship Source Object** to the AIF_HS_DRM_HIERARCHY_V view.
8. Click **Save**.

Configuring Domains

Domains are optional for the integration. They are needed when the dimensions from the source systems contain nodes of different types with the same name. An example would be both an account 1234 and a product 1234. Domains can make these nodes independent of each other with the use of the domain qualifier.

The Oracle Hyperion Financial Data Quality Management, Enterprise Edition integration passes in an Oracle Data Relationship Management domain qualifier in the staging tables used by the imports. This information can be used by the import and blender Data Relationship Management processes to use Data Relationship Management domains. The Data Relationship Management domain value to be used is specified in FDMEE.

The imports and blender in the FDMEE application template are preconfigured to work with domains.



Note:

After import, the nodes are not put into the domains. This can be done manually or as part of a blend process.

Creating Domains in Data Relationship Management

Create a Data Relationship Management domain for each value to be passed in from FDMEE.



Note:

The Data Relationship Management user needs to have the Application Administrator role.

To create a domain:

1. On the Data Relationship Management Home page, click **Administer**.
2. From **New**, select **Domain**.
3. Enter a name and description for the domain.
4. Enter a **Qualifier** and **Delimiter** as needed to prevent conflicts between dimensions.
5. Determine if there should be a suffix or prefix. Prefix is the default and most used form.
6. Click **Save**.

Manually Setting Domains for Imported Nodes

The nodes can be added to the appropriate domains after import.



Note:

The Data Relationship Management user needs to have the Application Administrator role.

To add nodes to domains after import:

1. On the Data Relationship Management Home page, click **Browse** and then click the version to work with.
2. Click the Property button to display Version Properties.
3. Set up the Version Property in the System Category “Domains in Use” with the domains to be used.

4. Click **Save**.
5. Right click the version and select **Domain Membership**.
6. Set the **Override Property** to the FDMEE.Node Domain Qualifier.
7. Click **OK**.

Using the Blender to Set Domains for Imported Nodes

When blending the nodes from the import version to the production version, the blender is set up to put the nodes into the domains for the target version.

Note:

If domains are set up to prevent conflicts of names, then the domains need to be configured in the version where the mapping is done as well as the versions to which the source system nodes were imported. Failure to do so will result in the suffix or prefix not being removed in the export that provides the mappings to FDMEE and will cause a mapping failure.

Configuring Blend After Import

The imports used by Oracle Hyperion Financial Data Quality Management, Enterprise Edition can be set up to perform a blend operation after a successful import. This allows the imported data to be moved automatically to a master version so that the mappings can be performed in Oracle Data Relationship Management.

Note:

The Data Relationship Management user needs to have the Application Administrator role.

To modify the import profile:

1. On the Data Relationship Management Home page, click **Import**.
2. Open the desired FDMEE import profile.
3. Click on the **Target** tab.
4. Select **Blend After Import**.
5. Set the Blender Profile to be used (FDMEE Blend).
6. Click **Edit Parameters**.
7. Add a Parameter with a Key of BLEND_TARGET and set its value to the version to be used. You can use the FDMEE_MAP_VERSION variable.
8. Click **OK**.
9. Click **Save**.

Extending Imports to Use Additional Custom Top Member and Aggregation

The Oracle Hyperion Financial Data Quality Management, Enterprise Edition staging tables for importing data to Oracle Data Relationship Management contain columns for 20 Custom Top Members and 20 Enable Custom Aggregation properties. The application template only provides properties for the first four.

The additional columns can be used by creating custom properties to hold the contents from the import. Create the properties with the same characteristics as the HFM.Custom1TopMember and HFM.EnableCustom1Agg properties and add them to the Node section of the FDMEE HFM import profile and map them to the appropriate column in the AIF_HS_DRM_MEMBER_V table.

Adding Mapping Validations

There are no predefined validations in this application template. This is due to the fact that the names for the EPM hierarchies are not fixed and the mapping validations are based on those hierarchy names.

Complete Mapping Check

This validation is intended to verify that all bottom (leaf) ERP nodes are mapped to an EPM hierarchy. The logic to fail the validation should be Core.Leaf = True AND Core.References does not contain EPM Hierarchy.

This can be done in multiple ways:

- Formula or script-based validation
- Query and a query-based validation
- Derived property and a property-based validation

Below is an example of a formula that could be used in a validation:

```
IF(AND(PropValue(core.leaf),Equals(integer,ArrayIndex(EPMHIERNAME,PropValue(Core.References),[comma]),0)),True,False)
```

Where EPMHIERNAME is the actual EPM hierarchy name.

Configuring the FDMEE Template for Custom Data Relationship Management Implementations

The Oracle Hyperion Financial Data Quality Management, Enterprise Edition application template is designed to work with the Performance Management Architect and Planning application templates.

Custom Oracle Data Relationship Management implementations supporting Oracle Essbase, Oracle Hyperion Financial Management, and Planning which do not use these application templates will have to modify the metadata from the FDMEE application template to use it for integration with FDMEE.

The changes that need to be made are in two main areas: imports and exports.

Imports

For imports, the Oracle Hyperion Financial Data Quality Management, Enterprise Edition system can send in default target application information for the nodes from the source application. If this portion of the integration is to be used, then the following imports must be modified to connect the database table to the appropriate custom Oracle Data Relationship Management property. If this functionality is not needed then the application-specific imports can be deleted (or not loaded) and only the FDMEE Import used.

FDMEE sends application-specific values in the Performance Management Architect format. For example the Data Storage Value will be “NeverShare” instead of “Never Share” and “ShareData” instead of “Shared.” This means that if the Custom Data Relationship Management implementation is set up using the Outline Utility values instead of the Performance Management Architect values then there will be a need to transform those values. This can be done by creating a lookup and using it to transform the values during import or to add a view onto the staging view that does the transformations.

FDMEE Essbase (ASO) Import

Map the following columns to the appropriate custom properties.

Import Section	Column	Description
Node	TIMEBALANCE	Essbase Time Balance
Node	VARIANCEREPORTING	Essbase Variance Reporting
Relation	DATASTORAGE	Essbase ASO Data Storage

FDMEE Essbase (BSO) Import

Map the following columns to the appropriate custom properties.

Import Section	Column	Description
Node	TIMEBALANCE	Essbase Time Balance
Node	VARIANCEREPORTING	Essbase Variance Reporting
Relation	DATASTORAGE	Essbase BSO Data Storage

FDMEE HFM Import

Map the following columns to the appropriate custom properties.

Import Section	Column	Description
Node	CONSOLIDATIONACCOUNTTYPE	Essbase Time Balance
Node	ISICP	Essbase Variance Reporting
Node	ISCONSOLIDATED	Essbase BSO Data Storage
Node	CUSTOM1TOPMEMBER	HFM Custom 1 Top Member
Node	CUSTOM2TOPMEMBER	HFM Custom 2 Top Member

Import Section	Column	Description
Node	CUSTOM3TOPMEMBER	HFM Custom 3 Top Member
Node	CUSTOM4TOPMEMBER	HFM Custom 4 Top Member
Node	ENABLECUSTOM1AGGR	HFM Enable Custom 1 Aggregation
Node	ENABLECUSTOM2AGGR	HFM Enable Custom 2 Aggregation
Node	ENABLECUSTOM3AGGR	HFM Enable Custom 3 Aggregation
Node	ENABLECUSTOM4AGGR	HFM Enable Custom 4 Aggregation

FDMEE Planning Import

Map the following columns to the appropriate custom properties.

Import Section	Column	Description
Node	ACCOUNTTYPE	Planning Account Type
Node	TIMEBALANCE	Planning Time Balance
Node	VARIANCEREPORTING	Planning Variance Reporting
Relation	MEMBERVALIDFORPLAN1	Planning Member Valid for Plan 1
Relation	MEMBERVALIDFORPLAN2	Planning Member Valid for Plan 2
Relation	MEMBERVALIDFORPLAN3	Planning Member Valid for Plan 3
Relation	MEMBERVALIDFORCAPEX	Planning Member Valid for Capex
Relation	MEMBERVALIDFORWORKFORCE	Planning Member Valid for Work Force
Relation	DATA STORAGE	Planning Data Storage
Relation	SOURCEPLANTYPE	Planning Source Plan Type

Exports

For each export there are two columns that will need to be mapped to the custom properties for the target dimension:

- Column: TARGKEY – The name of the node being mapped to in the target dimension.
- Column: TARGETDIMENSIONNAME – The dimension name of the target dimension.

The property used to get the Target Key Name cannot just use a formula of ParentProp(). This is because you may map at a rollup point from the ERP dimension. The logic for this property should continue up and return the first node name found that is a true target dimension node. For example:



Loading GL Values and Hierarchies into Data Relationship Management

Importing Members from FDMEE

The import process is run using an Oracle Hyperion Financial Data Quality Management, Enterprise Edition metadata rule and will populate the staging tables and execute the Oracle Data Relationship Management import to create the version within Data Relationship Management.

See *Oracle Hyperion Financial Data Quality Management Administrator's Guide*.

Blending ERP Members to EPM Versions

The blend profile **FDMEE Blend** is set up to blend the versions imported from FDMEE into existing Data Relationship Management versions. The blend can be run manually within Data Relationship Management or it can be run at the end of a FDMEE metadata rule that imports the data into Data Relationship Management. This can be done by configuring the Blend after Import functionality on the import profiles used for the FDMEE import process.

Note:

The Data Relationship Management user needs to have the Application Administrator role.

To blend the ERP members to EPM version:

1. On the Data Relationship Management Home page, click **Import**.
2. Open the import profile by double clicking on it.
3. Click on the **Target** tab.
4. Select the **Blend After Import** checkbox.
5. Select the appropriate blender (FDMEE Blend is the one supplied by the template) from the blender profile drop down list.
6. The Default parameters in the supplied FDMEE Imports set the `BLEND_TARGET` parameter to be the `FDMEE_MAP_VERSION` variable. If this is not the desired target of the blend then this parameter needs to be changed to point to the appropriate version or version variable. It is recommended that you use a version variable so that the import profiles do not have to be modified as the target version changes. To set the Parameter `BLEND_TARGET` do the following:

- a. Click the **Parameters** button (Edit Parameters).
 - b. Click on **Edit** button for the BLEND_TARGET row.
 - c. Set the **Value** to the appropriate version variable or version.
 - d. Click the **Save** button in the BLEND_TARGET row.
 - e. Click **OK** on the Parameter Edit dialog box.
7. Click the **Save** button to save the changes to the import profile

Mapping ERP to EPM Members in Data Relationship Management

Related Topics

- [Setting the FDMEE_MAP_VERSION Variable](#)
- [Performance Management Architect and Planning Dimensions Used for Mappings](#)
- [Mapping ERP to EPM Members in Data Relationship Management](#)
- [Exporting Mappings to FDMEE](#)

Setting the FDMEE_MAP_VERSION Variable

The Version Variable FDMEE_MAP_VERSION is used by FDMEE when running exports to extract mapping data from Oracle Data Relationship Management.

Before the Oracle Hyperion Financial Data Quality Management, Enterprise Edition system runs the mapping exports this variable needs to be assigned to the version that contains the mappings.



Note:

The Data Relationship Management user needs to have the Application Administrator role.

To set the FDMEE_MAP_VERSION variable:

1. On the Data Relationship Management Home page, select a version.
2. Right click on the version and select **Set Variables** and then **FDMEE_MAP_VERSION**.

Performance Management Architect and Planning Dimensions Used for Mappings

The imports from Oracle Hyperion Financial Data Quality Management, Enterprise Edition do not configure hierarchies in Oracle Data Relationship Management to be immediately used for managing mappings for a target EPM application. Data Relationship Management hierarchies must be separately configured for EPM dimensions using properties included in the Performance Management Architect or Oracle Hyperion Planning application templates for Data Relationship Management.

 **Note:**

Hierarchies should not be configured with a shared dimension for Performance Management Architect when used to manage member mappings for FDMEE. Hierarchies used to manage mappings for FDMEE should be configured as local dimensions only (using the Essbase/HFM/HP Dimension properties) to ensure the mappings can be appropriately validated against the target application by FDMEE.

At a minimum, the following properties need to be properly configured and populated for the FDMEE mapping export process to function correctly.

- Dimension
 - ESSBASE.Dimension
 - HFM.Dimension
 - HP.Dimension
- Membership
 - ESSBASE.Membership
 - HFM.Membership
 - HP.Membership
- MemberName
 - ESSBASE.MemberName
 - HFM.MemberName
 - HP.MemberName

Dimension (HFM, Essbase, HP)

Configure the following hierarchy level properties to add the appropriate values to the allowed values:

- HFM.Dimension
- HP.Dimension
- ESSBASE.Dimension

This property must be set to the appropriate value for the target Performance Management Architect or Planning hierarchies.

Membership

The top nodes of the Performance Management Architect and Planning hierarchies should have this property set to “Parent Member”. Bottom nodes of the Performance Management Architect and Planning dimensions will derive to “Member Node” by default. In the locations where mapped nodes are beneath the actual bottom nodes of the Performance Management Architect and Planning dimension nodes, the bottom Performance Management Architect and Planning dimension nodes will need to be set to “Base Member.”

MemberName

This property returns the name of the node if the node is either a “Parent Member” or “Base Member” if it is at the top of the hierarchy it returns a blank otherwise it rolls up the ancestor chain and returns the first node name that is either a “Parent Member” or “Base Member.” For this to work with the mapping exports the appropriate Membership properties have to be set.

Mapping ERP to EPM Members in Data Relationship Management

Oracle Data Relationship Management hierarchies are used to manage explicit member mappings from GL segment values of an ERP source system to dimension members of an EPM target application. Data Relationship Management hierarchy relationships can be used to manage one-to-one or many-to-one mappings between segment values and dimension members.

After the ERP hierarchies have been blended into the version that contains the EPM hierarchies, the nodes can be mapped by placing the ERP nodes under the EPM node to be mapped to. The ERP node can be either a bottom (leaf) node or a structure (limb) node. If a structure (limb) node is used then it will be equivalent to mapping all of the bottom (leaf) nodes under it. The mappings sent to Oracle Hyperion Financial Data Quality Management, Enterprise Edition will not include the ERP structure (limb) nodes.

There are multiple utilities that can be used in Data Relationship Management to assist with the mappings:

- Query
- Compare
- Action Script

Data Relationship Management queries can be used to identify ERP nodes that are not mapped to an EPM hierarchy by using either the Core.Links or Core.References statistical properties to see if it contains the EPM hierarchy. The results of the query can then be copied onto the clipboard. After the unmapped nodes are on the clipboard, they can be mapped to the EPM hierarchies by opening the appropriate EPM hierarchy and copying the ERP nodes from the clipboard onto the EPM hierarchy.

Compares can be used to compare the ERP nodes in both the ERP and EPM hierarchies to identify unmapped nodes. The nodes can then be mapped by drag and drop, via the clipboard, or by a manual insert.

In addition to using the Query and Compare features interactively in Data Relationship Management, the results can be exported to a spreadsheet. The mappings can be entered into the spreadsheet to generate an action script to perform the insert actions for the mappings.

Note:

To map an ERP node to the EPM node, the bottom node of the EPM hierarchy will need to be changed from a leaf node to a limb node and its appropriate membership property set to Base Member.

Following is an example HFM EPMA hierarchy with mapped nodes under the base membership nodes.

Name	HFM Membership	HFM Member	HFM Dimension	EPM Location Names	Domain Node Name
Geographical	Parent Member	Geographical	Entity		
ASIA	Parent Member	ASIA	Entity		
China	Parent Member	China	Entity		
Japan	Parent Member	Japan	Entity		
Singapore	Parent Member	Singapore	Entity		
SingaporeAdmin	Base Member	SingaporeAdmin	Entity	EBS_COMMA7DIM	28
ent_28	Not Applicable	SingaporeAdmin	Entity	PS_COMMA7DIM	YAMAGATA
ent_YAMAGATA	Not Applicable	SingaporeAdmin	Entity		
SingaporeProduction	Base Member	SingaporeProduction	Entity		
ent_01	Not Applicable	SingaporeProduction	Entity	EBS_COMMA7DIM	01
SingaporeSales	Base Member	SingaporeSales	Entity		
Canada	Base Member	Canada	Entity		
ent_CANADA	Not Applicable	Canada	Entity	PS_COMMA7DIM	CANADA
ent_ALBERTA	Not Applicable	Canada	Entity	PS_COMMA7DIM	ALBERTA
ent_NEWBRUNS	Not Applicable	Canada	Entity	PS_COMMA7DIM	NEWBRUNS
ent_NOVASCOT	Not Applicable	Canada	Entity	PS_COMMA7DIM	NOVASCOT
ent_QUEBEC	Not Applicable	Canada	Entity	PS_COMMA7DIM	QUEBEC
ent_TORONTO	Not Applicable	Canada	Entity	PS_COMMA7DIM	TORONTO
ent_YUKON	Not Applicable	Canada	Entity	PS_COMMA7DIM	YUKON
Europe	Parent Member	Europe	Entity		
JointVenture	Parent Member	JointVenture	Entity		
SouthAmerica	Parent Member	SouthAmerica	Entity		
UnitedStates	Parent Member	UnitedStates	Entity		

In this example, ent_28, ent_YAMAGATA, ent_01 are bottom level source system nodes mapped under bottom level target system nodes. Note that the target system nodes (Singapore Admin and Singapore Production) are no longer leaf nodes and have the Membership property set to “Base Member.”

Ent_Canada is an example of mapping a limb node from the source hierarchy to base member in the target hierarchy. As before the base member in the target is now a leaf node with the membership set to “Base Member.”

Bottom Level Mapping Only

It may be desirable to enforce that mapped nodes are mapped to a bottom level EPM node. This can be done by checking if the node type (EPMA.MemberType) is FDMEE ERP and if the Parent is either a node with an FDMEE ERP node type or, if not, that it is a Base member of the EPM hierarchy (HFM.Membership, Essbase.Membership, HP.Membership, EPMA.Membership).

Exporting Mappings to FDMEE

The export process is run from Oracle Hyperion Financial Data Quality Management, Enterprise Edition using the Import Mappings from Oracle Data Relationship Management option and will execute the Data Relationship Management export to populate the staging tables for FDMEE to process. The Version Variable FDMEE_MAP_VERSION must be set to point to the appropriate version where mappings for FDMEE are being managed.

See *Oracle Hyperion Financial Data Quality Management Administrator's Guide*.